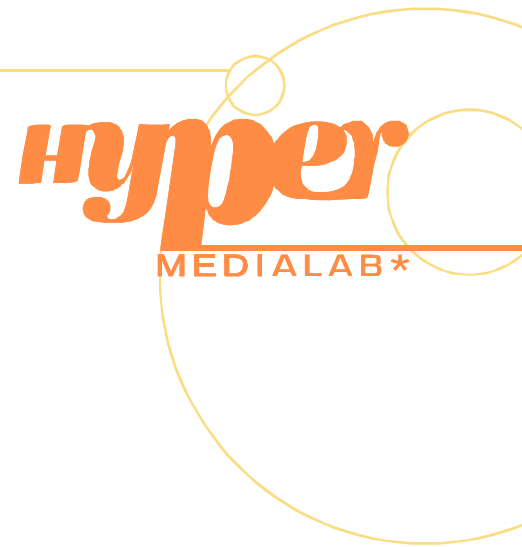




Modding scenes

Introduction to user-created
content in computer gaming

Tero Laukkanen



Modding scenes

Introduction to user-created content in computer gaming

Featuring case studies of "*Half-Life*", "*The Sims*" & "*Grand Theft Auto III / Vice City*" modding scenes

Tero Laukkanen

tero.laukkanen@uta.fi

October 2005

University of Tampere Hypermedia Laboratory

<http://www.uta.fi/hyper/>

Table of contents

1. Executive summary	4
2. Introduction	5
2.1. Modding as participatory culture	5
2.2. The evolution of the modding culture	7
2.3. About the research	14
2.3.1. About terminology	15
2.3.2. Release note (about releases)	17
3. Case study: Half-Life	18
3.1. Description of the game	18
3.2. Overview of the modding scene	20
3.2.1. Organization of the modding scene	21
3.3. User-created game content	30
3.3.1. Maps	32
3.3.2. Textures	36
3.3.3. Prefabs	38
3.3.4. Map models	40
3.3.5. Custom characters (models and skins)	42
3.3.6. Weapon models & skins	45
3.3.7. Sprites	47
3.3.8. Logos	48
3.3.9. Sounds	48
3.3.10. Code modifications	49
3.3.11. Total conversions	52
Examples: Multiplayer	55
Examples: Single player	57
3.4. Summary and notes	59
3.4.1. Notes	61
4. Case Study: The Sims	64
4.1. Description of the game	64
4.2. Overview of the modding scene	66
4.2.1. Organization of the modding scene	68
4.3. User-created content	73
4.3.1. Skins	75
4.3.2. Meshes	79
4.3.3. Objects	81
4.3.4. Hacked objects	86
4.3.5. Walls, floors and roofs	90
4.3.6. Lots	91
4.3.7. Tools	94
4.4. Summary and notes	96
4.4.1. Notes	98
5. Case Study: Grand Theft Auto III & Vice City	101
5.1. Description of the game(s)	101

5.2. Overview of the modding scene	103
5.2.1. Organization of the modding scene	104
5.3. User-created content	109
5.3.1. Map add-ons	110
5.3.2. Textures	114
5.3.3. Player models and skins	116
5.3.4. Vehicle models and textures	119
5.3.5. Weapon models	122
5.3.6. Code modifications	124
5.3.7. Major modifications	127
5.3.8. Modifiers	132
5.4. Summary and notes	133
5.4.1. Notes	134
6. Summary	137
6.1. Comparing the cases	137
6.1.1. Organization of the modding scene	138
6.1.2. Forms of custom content	140
6.1.3. Tools and support	144
6.2. Further research objectives	148
7. References	150

1. Executive summary

“Modding” is an increasingly popular participatory practice where game fans modify and extend officially released game titles with their own creations. This report introduces the reader to the diversity of user-created game content and to the multifaceted online networks – referred to here as “modding scenes” – employed in the making and sharing of the “mods”.

After a brief introduction to the modding culture in general and its evolution from a shady subculture to a viable marketing strategy, the modding scenes of three games are examined in more detail. The games are *Half-Life* (Valve 1998), *The Sims* (EA/Maxis 2000) and *Grand Theft Auto III / Vice City* (Rockstar 2003/2003). Of the three, *Half-Life* represents first-person shooters, which have traditionally been at the forefront of modding. *Half-Life* itself is arguably the most modified game of all time. *The Sims*, often described as a virtual dollhouse, has been an unprecedented crossover success, and due to the developer's conscious efforts to simplify custom content creation, has also greatly expanded the demographic profile of modders. The third game under examination is actually two games; *Grand Theft Auto III* and its sequel *Grand Theft Auto: Vice City* share a common modding scene. What makes the *GTA* scene particularly interesting is the fact that it has blossomed without any support from the developer.

The modding scenes of the three games are presented as separate case studies. Each case study begins with a brief introduction of the game itself to give the subsequent discussion some context. The organization of the modding scenes – in essence the role of various online networks – and developer involvement is discussed next. The greater part of each case study is devoted to a detailed analysis of the various forms of user-created content for the particular game. In addition to the custom game content itself, the tools modders employ in the creation of the custom content are examined. Each case is concluded with a brief summary followed by specific insights gained from studying the particular modding scene. While the case studies themselves include some references to each other, more thorough comparisons are made in the overall summary that wraps up the report. The summary also presents some general concepts for systematic review of modding scenes.

The approach of this report is primarily descriptive. The aim is to provide a comprehensive empirical groundwork for further, more focused and analytical approaches into the modding phenomenon.

The report was produced in the TEKES funded Mobile Content Communities (MC2) research project.

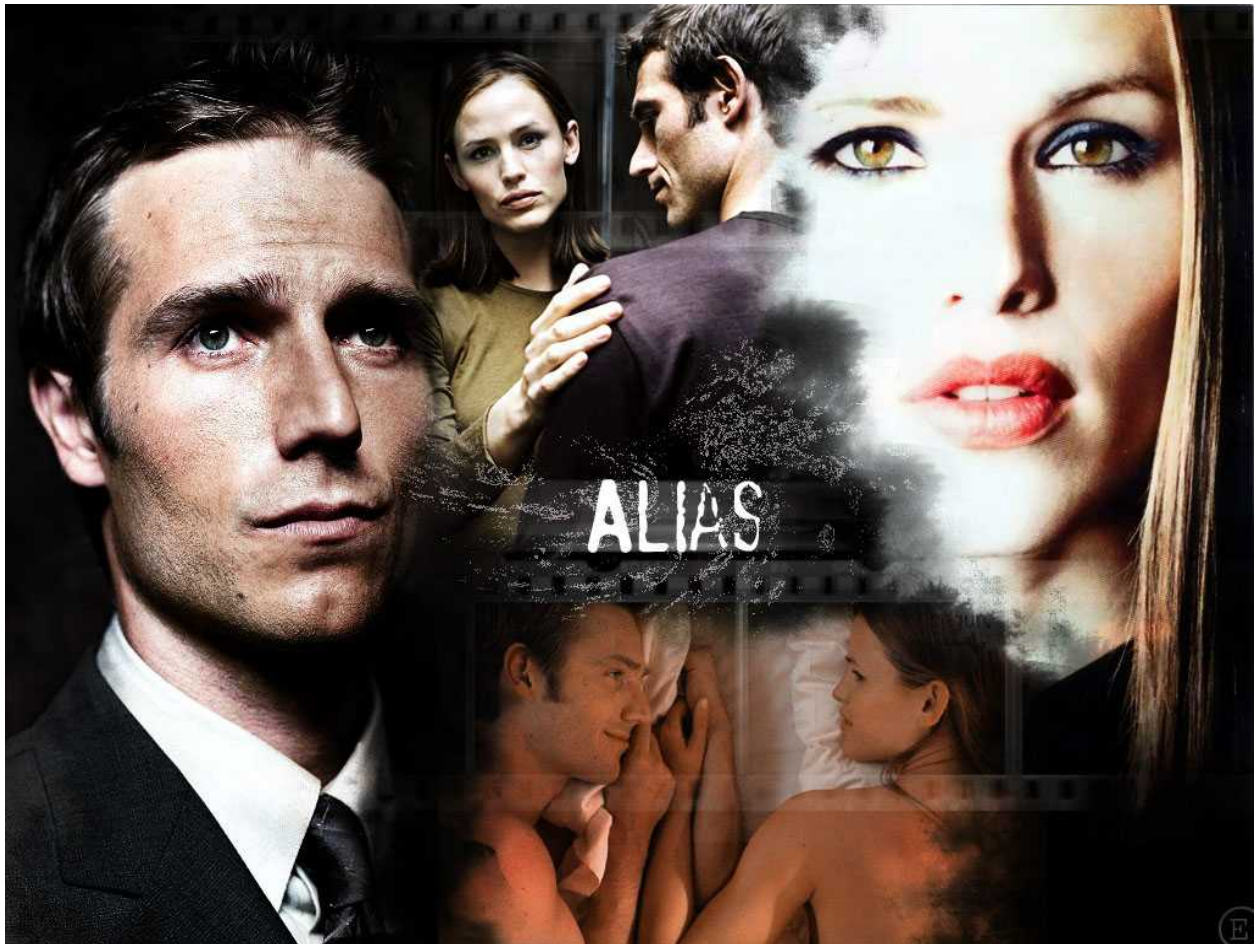
2. Introduction

2.1. Modding as participatory culture

The practice of “modding” - i.e. modifying and extending officially released games with fan-produced content - is arguably one of the most distinctive features in current computer gaming culture. From the somewhat dubious sub-cultural beginnings modding has matured into the mainstream during the past decade. Increasing amounts of fan creations, ranging from skins (graphics that change game character’s clothing/appearance) to total conversions (new games build on existing game engines), are downloaded from the Internet, constantly breathing fresh life to aging titles. Not surprisingly, gaming industry has taken notice. Inspired by the commercial success and, especially, longevity of several “mod-friendly” titles, a growing number of developers now perceive user-created content as an important part of their marketing strategies (Edge 2003, 58; Equip PC 2003, sidebars). Consequently fan production is supported in multiple ways: official editing tools are bundled with the games, source codes opened to public, support forums and information resources hosted on the web, in-house workshops and seminars organized, high-profile modding contests sponsored, and so on. The industry is not interested in modding just because the availability of free additional content can sell more games. Developers also recognize the more implicit long-term value of the goodwill generated among the hard-core audience. In addition, the modding scenes are commonly acknowledged as one of the most important breeding grounds for new talent, and as areas where innovative concepts can still emerge and be tested without financial risks (Equip 2003, 73; Au 2002).

Recently modding has also attracted the attention of the academics, especially those with background in cultural studies. With its collaborative DIY aspects, appropriative nature and innovative use of new technology, modding represents a particularly illustrative example of the so-called “participatory culture”. Participatory culture - a term coined by Henry Jenkins in his seminal study of fan cultures (Jenkins 1992) - refers to a form of media consumption where audiences take an especially active role in the circulation of media texts, effectively blurring the boundaries between audiences and producers. Though the concept of participatory culture is not new, rapid advances in digital media technologies during the last decade have given audiences increasingly powerful tools for grass-roots production and publishing. Even more importantly, Internet has provided an exceptional platform for informal support communities and distribution networks needed to sustain this kind of

productivity. Without question the digitalization of media environment has been elemental in the diffusion of the practices of participatory culture (Jenkins 2003; see also Laukkanen 2004). From the perspective of cultural scholars, then, the interesting thing about modding is not how profitable it is for the gaming industry, but how it challenges the traditional cultural hegemony and allows more diverse voices to participate in the shaping of our cultural heritage.



Modders are not unlike media fans, who use officially released texts as the basis of their own cultural creations. "Traditional" fans have also embraced digital technologies, as this desktop wallpaper collage from a fan of the TV series *Alias* illustrates.

Of course, this kind of fan productivity is not particular only to gamers; other media fans have for a long time been known to produce stories ("fan fiction"), drawings, songs, videos, web sites, and so on inspired by their favorite TV shows and movies (for an in-depth account see Jenkins 1992). However, because of their inherently digital and (inter)active nature games seem to provide an especially suitable platform for consumer participation. Furthermore, fan communities are practically built-in to online multiplayer games. Gaming industry is also the only sector of entertainment industry that has been able (or willing) to form a mutually beneficial alliance with fan producers. Whereas TV and movie studios have become notorious for shutting down Internet fan sites on account of

copyright violations, the gaming industry has taken the opposite route, actively encouraging and supporting fan production (Jenkins 2002, 165-166). Naturally game developers have also set certain boundaries to fan producers, but policing of these boundaries has (for the most part) been based on trust, not on fear.

While the positive aspects of consumer participation are often recognized and accepted, it should also be noted that there are people in the gaming industry who feel that the money spent on developing editing tools and supporting web communities for the niche audience of hard-core gamers could be more wisely spent elsewhere (Edge 2003, 71-72). Equally, some academics are quick to point out the obstacles and limitations fan producers such as modders have to deal with, and the conniving ways in which the industry regulates and co-opts fan production for its own capitalistic purposes (Mactavish 2003, Kücklich 2005; see also Sotamaa 2003, 14-16). These contradictory perspectives naturally make modding all the more interesting and worthy area of investigation.

2.2. The evolution of the modding culture

While computer game modding is, naturally, quite a recent phenomenon, the practice of modifying games has probably been around just as long as games themselves have. Chess, for example, has inspired hundreds of variations during the centuries. Some of them have been drastic alterations, changing everything from the board up, some just minor rule changes. Some have been quickly forgotten, some have survived, and some have gained such popularity they have eventually become the current "official" version of chess. At a more mundane level, most of us probably have played board games such as Monopoly with slight adjustments to the rules (e.g. money can be loaned from other players).

The exact origins of computer game modding are also difficult to pin down. Partly this is because there is very little historical documentation on the subject, and partly because it is debatable what exactly constitutes 'modding'. It might again be justified to say that computer game modifications are just as old as computer games themselves. In fact, the first-ever computer game, Spacewar, distributed from 1962 onwards as a demo/testing software on a room-sized minicomputer, was enthusiastically modified in practically every university campus where the minicomputer itself was installed¹. While it would be quite a stretch to call the handful of Spacewar hackers on selected research facilities a modding culture, they nevertheless

¹ <http://www3.sympatico.ca/maury/games/space/spacewar.html>

were doing the same thing that modders do today: modifying a game someone else has created to their own personal likings. Furthermore, the pioneering hackers exhibited certain qualities, which would become crucial not only in the development of modding culture, but also in the overall advancement of information technology. These qualities are often summed up as the 'hacker ethic'. As Steven Levy reflects on the popularity of Spacewar hacking:

"The group effort that stage by stage had improved the program could have stood for an argument for Hacker Ethic: an urge to get inside the workings of the thing and make it better had let to measurable improvements. And of course it was all a huge amount of fun." (Levy 1994, 65)

Hackers were driven by their passion to examine, understand and ultimately improve on someone else's code. Although that passion occasionally came at the expense of healthy social life, hacking itself was very much socially motivated (Himanen 2001, 57). Instead of money, the ultimate reward for hackers was the respect of their peers. In the early 1960s these peers were colleagues and fellow students at the computer labs, in today's modding scenes they are the fellow members of global online communities.

Modding could not develop into a more widespread hobby until home computers with decent gaming and programming capabilities became commonplace. To some extent this happened already in the late 70s to early 80s with machines like Apple II and Commodore 64. The latter especially became an extremely popular gaming platform for many years. The popularity of the early home computers gave birth to a cracker subculture that - although operating from the "dark side" of the hacker ethic - could be seen as some sort of a guidepost in the evolution of the current modding culture. Crackers were hardcore computer hobbyists who used their programming skills to enable gameplay cheats (such as infinite lives) and to remove the copy protection schemes of commercial games. "Cracked" games were then distributed freely among the growing subculture. At some point it became customary for crackers (or cracker groups) to include personalized intro/load screens to the cracked games. These intros were like electronic graffiti; they were used to build a reputation among the subculture and to show off the crackers' programming skills.² First intros (a.k.a. "cracktros") were simple text screens but gradually the graphics became more complex and animated, and self-composed music became an important feature. Before long the intros themselves became the focal point for some crackers, and eventually

² <http://www.scheib.net/play/demos/what/hugi/>

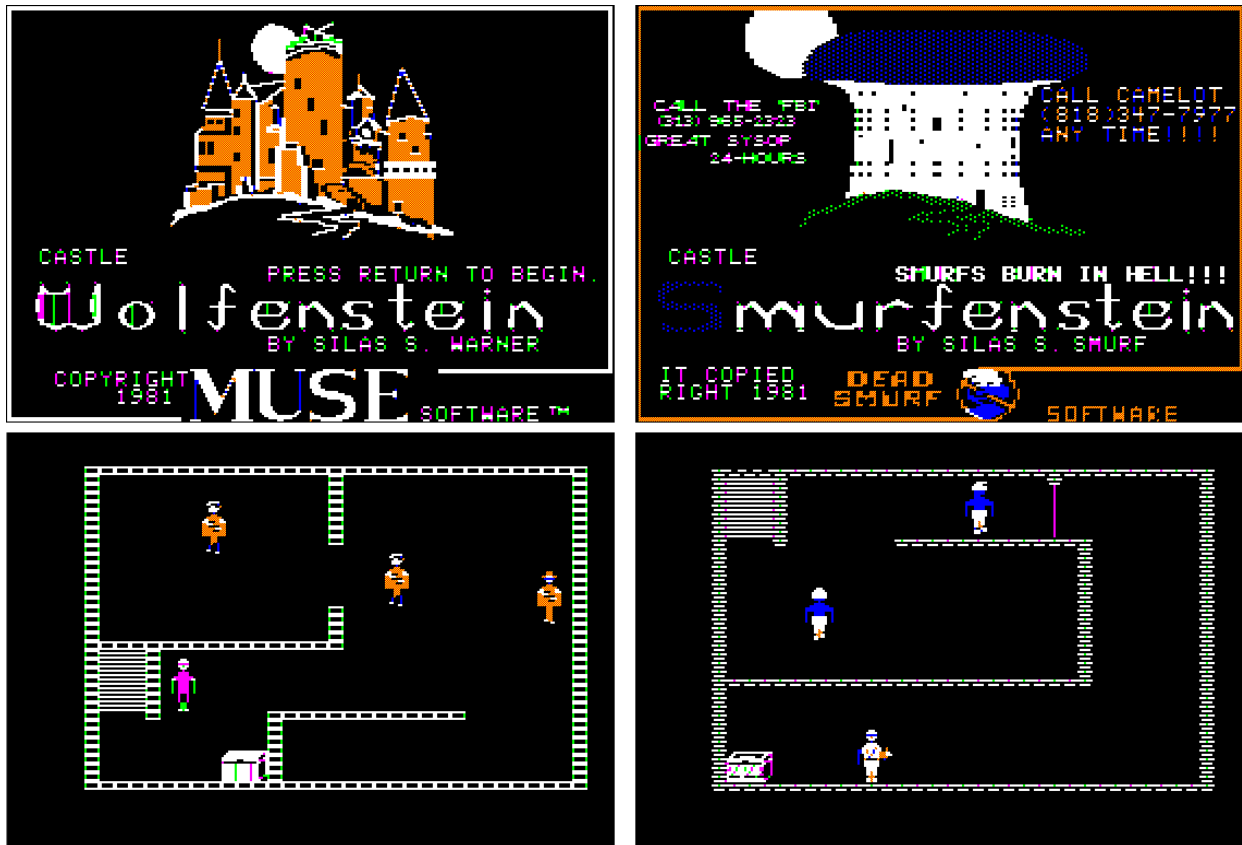
spawned a still active computer subculture of their own known as the “demo scene”³.



Customized intro screens (“cracktros”) were a way for cracker teams to gain notoriety among the subculture. The above screenshots are from the cracktros of two different groups for the C-64 game Nobby the Aardvark. (Source: <http://www.freax.hu/sample.html>)

The cracker intros did not change the actual game, so strictly speaking they were not really mods as understood in this report. Furthermore, the cracker (or “warez”) subculture still exists as a distinctly different subculture from the now legitimate modders. However, some inspired crackers of the era did more than just add new intro screens. A famous example of such creative undercurrent was *Castle Smurfenstein*. Created by two American high school students in 1983, this parody version of an Apple II game *Castle Wolfenstein* not only included a custom-made intro screen, but also replaced the game’s nazi characters with those of the popular 80’s cartoon *The Smurfs*.

³ http://en.wikipedia.org/wiki/Demo_scene



Castle Smurfenstein, a 1983 modification of the Apple II game Castle Wolfenstein, had the player battling against killer Smurfs instead of Nazis. Screenshots are from an Apple II PC emulator.

Although often cited as a pioneering example of modding culture, Castle Smurfenstein was nevertheless just as unauthorized as the warez distributed by crackers⁴. The birth of a legitimate modding would have to wait another decade, until the arrival of id Software's revolutionary demon kill-fest *Doom* (1993). Id's lead programmer John Carmack held the hacker ethic close to heart - only a few years earlier he had been just an enthusiastic hacker himself. Inspired by the innovative, yet unauthorized, modifications of id's previous title *Wolfenstein 3D* (1992) he made a conscious effort to design *Doom* in a way that would facilitate custom content creation. (Kushner 2003, 165-169) In practice this meant separating the art assets, such as level architecture, graphics and audio, from the main program. This content was stored in 'WAD' files, which could be edited without touching the program code and - crucially - shared independently from the executable.⁵ Carmack's novel design

⁴ Although fondly remembered, Castle Smurfenstein was definitely not *the* first mod. Even the makers of Castle Smurfenstein had already created another mod in the Smurf-theme called Dino Smurf (a parody of another Apple II game called Dino Eggs). <<http://www.evl.uic.edu/aej/smurf.html>>

⁵ WAD was an acronym for "Where's All the Data?" (Kushner 2003, 166). *Doom*'s successor *Quake* also used WADs. Because of id's games' pioneering

solution provided a technical foundation for the first legitimate modding scene. Of course, now that a commercial party was involved in the scene, there were bound to be some restrictions to hacker freedom. Id insisted that the fan-made WADs should only work with the full retail version of *Doom*, not the freely available demo version. Demonstrating a distinctly different mindset from crackers, most modders respected id's wishes. As a consequence of id's and modders' pact anyone who wanted to try the custom WADs online communities were raving about had to buy the retail version of *Doom*. This naturally boosted the already impressive sales of the game and set a precedent for the mutually beneficial relationship between game developers and their fans.



Unarguably one of the most influential computer games of all time, id Software's *Doom* (1993) was also the game that instigated the legitimate modding culture. Fan-produced extra levels and impressive "total conversions" such as *Alien TC* (pictured on the right) shared over the rapidly expanding Internet were an integral part of *Doom*'s success.

Without undermining Carmack's and id's influence, the fact that *Doom* would become the first game to spawn an active modding scene was also very much due to timing. At the time when *Doom* became a hit, the Internet was really starting to take off thanks to the advent of World Wide Web. Were it not for the rapidly expanding connectivity and blossoming online communities, the gospel of *Doom* WADs would not have reached such masses and the genesis of modding would have been further postponed. Id's initial support for modding was not particularly extensive. For example, it was up to the modders themselves to build the specialized tools required for editing the WAD file content. (Carmack had only left the files un-encrypted and open for exploration.) Using the online forums to pool their knowledge modders were able to build on each other's findings and shortly after the game's release first "unofficial" editors were already available (Antoniades 1994, 39-41).

Encouraged by *Doom*'s success the developers of next generation FPS games took their supportive efforts to increasing heights. Games such as id's own *Quake* (1996), Valve Software's

role in the modding culture, for many early modders 'WAD' was/is synonymous with 'mod'.

Half-Life (1998) and Epic Games' *Unreal Tournament* (1999) were designed to be even more "modder-friendly", official modding tools were bundled on the game CDs, online communities set up, and even in-house workshops organized for modders. The viability of modder support was affirmed in a spectacular manner when Counter-Strike, a team-based multiplayer total conversion for *Half-Life* became an online phenomenon at the turn of the millennium, attracting more players than all the professionally produced FPS titles (such as *Quake III: Arena* (id 1999) and *Unreal Tournament*) combined. In 2001 Counter-Strike became the first fan-made modification to be released commercially, selling millions even though it was still freely available for download. Five years later Counter-Strike is still the most popular online action game.⁶ Although other FPS developers have so far not been able to replicate the Counter-Strike phenomenon, they have nevertheless benefited enormously from their alliance with the modding community. Original total conversions catering for diverse tastes and the steady stream of user-created maps have become essential for the longevity and committed fan base for online games.⁷ And although not all gamers are interested in mods themselves, a lively modding scene also generates valuable kudos in the larger gaming community. In short, it would be more or less a commercial suicide to release an FPS game for the PC platform without any support for modding.

Although FPS games have traditionally been the pioneers and poster boys for the modding culture, user creativity has been embraced in other genres as well. BioWare's marketing for its online/off-line role-playing game *Neverwinter Nights* (2002), for example, focused on the bundled Aurora toolset, which enabled users to create their own scenarios much in the vein of traditional tabletop role-playing (Edge 2003, 58; see also Equip PC 2003, 68/sidebar). Of course *The Sims* (EA/Maxis 2000), examined in this study, is as far removed from FPS games as possible, yet relies heavily on user-created content. It has also introduced modding to a new breed of "casual modders". Other genres where modding is quite commonly supported include simulators (see Banks 2003), strategy games and sports titles. According to a survey conducted in 2002, more than one third of developers provided modding tools for their games. (Jeppesen

⁶ While there are no statistics of all online gaming activity, the statistics provided by the largest online gaming "hub" Gamespy (<http://archive.gamespy.com/stats/>) are a good indication of the games' general popularity.

⁷ Unlike single-player games where players usually advance from level to level in a predetermined order, online multiplayer battles are always fought on a single level, or a map. Once the battle is over, a new map is loaded and the fighting continues. Server providers can choose which maps to offer. While the developer usually provides 10-20 official maps with the game, user-created maps can just as easily be added to the server's map rotation.

and Molin 2003, 7) Since then the percentage has most likely still increased. Even whole gaming concepts, such as the massively multiplayer online game *Second Life*⁸ or the Sims' lead designer Will Wright's upcoming *Spore*⁹, are now based upon on user-creativity.

Thus far the legitimate modding culture has concentrated on computer gaming. Although many console games have featured in-game content editors, sharing of the custom content has been problematic. However, with the increased connectivity and memory capacity of consoles it seems that the PC exclusivity is soon coming to an end. (Edge 2003, 65) Microsoft, for example, has announced that the users of its next-generation console Xbox 360 could share and possibly even sell their creations over the Xbox Marketplace online service alongside professional content providers.¹⁰

As computing power has increased and game technology grown more complex, modding has also become more demanding. Whereas a modder with some experience could create a new level for *Doom* in just few hours, it can take weeks to create one for a current game such as *Half-Life 2* (Valve 2004). Total conversion mods are already created by development teams with up to dozens of volunteers specializing in different aspects of modding - modelling, texturing, level design, sound design, voice acting, programming, etc. Even with these large teams mods often take a year or two to reach a public beta testing phase, since amateur modders are also bound by "real-life" duties such as school or work. It does seem likely, as has been predicted, that in the near future mod teams will need to start acting more and more like professional development teams to get anything done at all (see Edge 2003, 64). Some have argued that the increased professionalism in modding could stifle creativity and lead to risk-aversion just as it has done in professional games industry (Kücklich 2005).

That said, it is also important to remember that user-created content comes in various forms - some are ambitious and complex, others small and simple. So while it might be true that modding for the state-of-the-art FPS games will require an increasingly professional approach in the future, as the cross-over success of *The Sims* modding has proven, there is still room for more casual fan production at the more accessible end of the scale.

⁸ <http://secondlife.com/>

⁹ <http://spore.ea.com/>

¹⁰ http://www.gamespot.com/news/2005/05/12/news_6124287.html

2.3. About the research

This study on computer game modding has been conducted under the TEKES funded Mobile Content Communities (MC2) research project. The multidisciplinary project has diverse research interests relating to mobility, communality, user experience and collaborative content production in gaming. One of the practical objectives is to develop open source tools and community services that facilitate user-creativity in the mobile context. While creative mobile gaming communities are still in the very early stages of evolution, many of the social and cultural practices projected to them have already been well established in the computer game modding scenes. As such, the study of modding scenes should provide valuable insights when developing the supportive practices and technologies for the mobile environment. In the context of the MC2 project, the purpose of this particular report is to serve as an introduction to the diversity of user-created content in computer gaming, and to provide a comprehensive empirical groundwork for further, more focused and analytical approaches into the modding phenomenon.

Since the intention of this study is to accumulate a thorough overview, it would be inadequate to focus on the modding scene of just one game. On the other hand, it would also be impossible to conduct any sensible research on all the games that are modified by users. As a manageable compromise, three particularly heavily modified titles have been chosen for in-depth case studies. The three games selected are the first-person-shooter *Half-Life* (Valve Software 1998), the driving-action game(s) *Grand Theft Auto III* & *Vice City* (Rockstar Games 2002, 2003), and the unique "people simulator" *The Sims* (EA/Maxis 2000). While three games can hardly account for the diversity of the entire modding culture, the selected titles provide a nice overview, as they all represent different game genres, and their modding scenes have evolved very differently. *Half-Life* represents the first-person shooters (FPS), which have traditionally been at the forefront of modding both in terms of popularity and developer support. *Half-Life* itself is perhaps the most modified game of all time, and its developer Valve has benefited enormously from its extensive modder support. *The Sims* has been an unprecedented crossover success as a game (though some might argue that the virtual dollhouse is not really a game), and due to the developer's conscious efforts to simplify custom content creation, has also greatly expanded the demographic profile of modders. The third game under investigation is actually two games; *Grand Theft Auto III* and its sequel *Grand Theft Auto: Vice City* share a common modding scene and could not really have been examined separately. What makes the *GTA* scene particularly interesting is that,

conversely to the *Half-Life* and *Sims* scenes, it has blossomed without any support from the developer.

The research method applied in this study has been practice-oriented. Numerous modding community web sites and discussion forums for each game have been explored, and a wide variety of user-created content downloaded and tested. Alongside custom game content various modding tools were also tried out. The systematic inspection of the three scenes is supported with more casual observation of modding culture in general.

The case studies will be presented with the following structure:

Description of the game

- Introduces the game.

Overview of the modding scene

- Contains a brief look at the game's modding scene, covering such issues as community structure and developer support.

User-created game content

- Discusses the various forms of custom content created for the particular game, as well as the official and unofficial modding tools applied in the creative process.

Summary and notes

- Brief summary of the findings followed by specific insights gained from studying the modding scene of the particular game.

Although the case studies can for the most part be read independently, some references and comparisons are made between the games.

2.3.1. About terminology

In the context of this report "modding" refers to the participatory practice where amateurs modify and extend commercially released games with their own creations. Accordingly, a "modder" is the amateur hobbyist participating in the craft of modding. In the processes of learning the craft, creating custom content and sharing their creations with other gamers, modders employ the resources of multifaceted collaborative online networks. Some of these networks are purely utilitarian, some of them more social in nature. Together they form "modding scenes". Each game has its own modding scene, but modders are by no means confined to modding just one game. The skills learned modding one game are often transferable to modding other games as well.

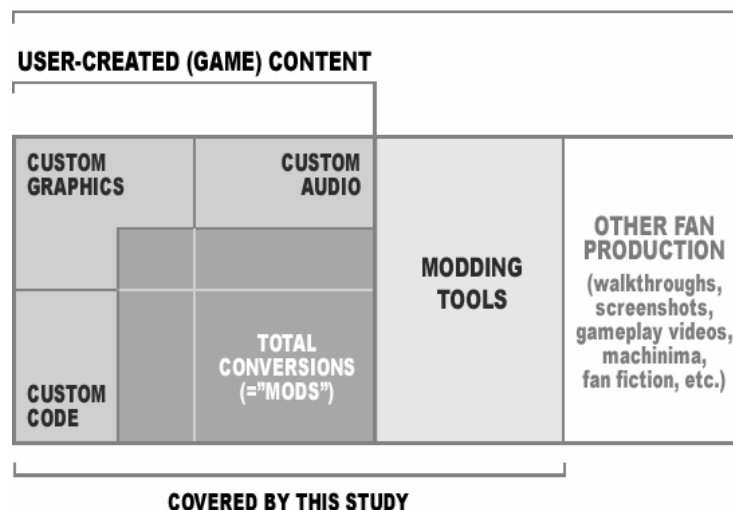
The term "mod/modification" often has a more specific meaning than just any fan-produced tweak or add-on. In many modding scenes the term is reserved specifically for the more

complex and thorough reworkings that are practically new games. The aforementioned *Counter-Strike*, for example, is a *Half-Life* mod/modification. Though *Counter-Strike* utilizes the original *Half-Life* engine, it changes everything else from the graphics to the game mechanics. Mods like *Counter-Strike* are also often called “total conversions” to emphasize the extent of customization. Somewhat confusingly, the word “mods” is also often used very loosely in reference to all modifications. In this report the term “user-created content” is preferred when talking about modifications in general. Different forms of custom content will be referred to by their specific names such as “skins”, “models”, “textures”, etc. Since the custom content types vary somewhat among the modding scenes investigated, these will be introduced separately for each case.

Finally, it must be noted that modding is just one form of game-related fan production. Game fans also produce and share different kinds of memorabilia, walkthroughs, gameplay videos, fan fiction and so on. These other forms of fan production will not be explored in this study.

While it is unfortunate and potentially confusing that the “lexicon of modding” is not yet fully established, the following diagram should clarify how some of the concepts used in this report relate to each other.

GAME-RELATED FAN PRODUCTION



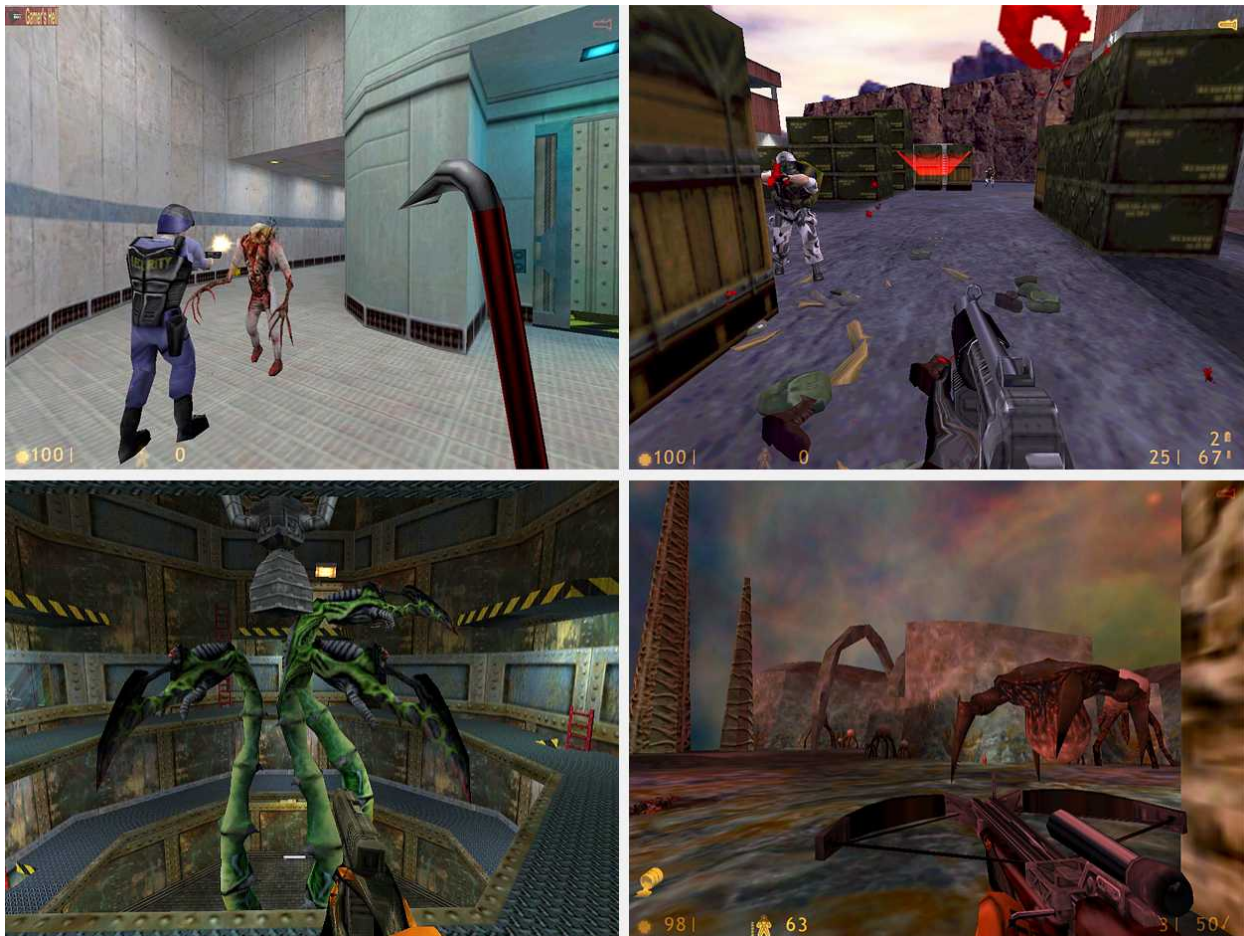
2.3.2. Release note (about releases)

This report was originally released internally for the MC2 research group in the fall of 2004. Since the original version was written, sequels have been released for all three games under examination. *Half-Life 2* (Valve) and *The Sims 2* (EA/Maxis) were released in the fall of 2004, *Grand Theft Auto: San Andreas* (Rockstar Games) for PC in the summer of 2005. The case studies have not been edited for this version, so any changes the sequels may have brought upon the corresponding modding scenes will not be discussed in the following.

3. Case study: Half-Life

3.1. Description of the game

At its core, *Half-Life* (Valve, 1998) is a traditional first-person-shooter (FPS) with a formulaic sci-fi premise. Player steps in to the shoes of Gordon Freeman, a military scientist whose working day takes an unpleasant turn when an experiment at the Black Mesa research facility goes awry. A portal opens to another dimension and soon the whole facility is infested with menacing mutants and aliens. Freeman must fight his way out of the underground facility, teleport to the other dimension and destroy the alien energy source. To make matters worse, heavily armed military units are deployed to cover up the accident and kill everybody at the contaminated facility, including Freeman. Luckily Freeman himself also comes across increasingly powerful human and alien weaponry (starting with the trademark crowbar). Occasionally the player will get AI controlled assistance from security guards and fellow scientists.



Half-Life vowed gamers and critics alike with its engrossing presentation. The varied level design, convincing enemy AI, occasional assistance from the NPCs, and numerous memorable set-pieces were among the elements that helped to create an unforgettable gaming experience.

Although at the surface *Half-Life* may seem formulaic, very few computer games in history have garnered as much appraisal. Released in 1998 it went on to win over 50 game-of-the-year awards¹¹, and quickly became a benchmark against which all future single-player FPS-games were to be measured. Built on a heavily modified *Quake* engine, *Half-Life* was technically impressive, but what ultimately secured its status as a classic was the exceptionally engrossing presentation. Everything from the minimal load times between levels and the placement of supplies to the numerous scripted sequences carefully woven to the storyline supported the player's immersion. For once critical accolades were reflected in the commercial success, and *Half-Life* became one of the best-selling computer games of all time.

In addition to the single-player campaign, *Half-Life* included basic multiplayer modes (deathmatch and team deathmatch). The only-adequate multiplayer capabilities were commonly regarded to be the low-point of an otherwise groundbreaking title. To remedy this Valve released a very successful free multiplayer add-on *Team Fortress Classic* (1999) less than a year after *Half-Life*.¹² TFC was also included in subsequent *Half-Life* update patches.

During the long wait for a proper sequel - *Half-Life 2* is currently scheduled to be released in the fall of 2004 after a year-long delay caused by stolen source code - several official add-ons using the original *Half-Life* engine were released commercially as stand-alone titles. The official add-ons have come from third party contractors, not from Valve itself. Of these, the single-player expansion packs *Opposing Force* (1999) and *Blueshift* (2001) were developed for commercial release from the outset, while the multiplayer add-ons *Counter-Strike* (2000) and *Day of Defeat* (2003) were already very successful, freely available mods when Valve released them commercially. *Gunman Chronicles* (2000), another single-player expansion, started out as an amateur mod project but Valve acquired it in the middle of development. The last official release *Counter-Strike: Condition Zero* (2004) was a commercially developed single-player version of the popular multiplayer mod. The commercial and critical success of the add-ons has been varied. The single-player add-ons especially have been quite poorly received. Most of the official add-ons have later been bundled to various special edition releases.

¹¹ This number comes from the case sleeve of *Half-Life Generation* bundle (Valve 2002).

¹² *Team Fortress Classic* was a recreation of an unofficial *Quake* mod *Team Fortress*, released in 1996. The modding team behind the original *TF* was hired/bought by Valve to create *TFC*.

The original *Half-Life* was released for the Playstation 2 console in 2002, and *Counter-Strike* for the Xbox console in 2003.

3.2. Overview of the modding scene

Considering that *Half-Life* itself utilizes a modified *Quake* engine, that the game levels were constructed with a tool that was originally an unofficial *Quake* modding tool, and that many of the Valve employees had background in the *Quake* modding scene, it is not all that surprising that Valve Software has been a keen supporter of fan productivity. *Half-Life* was shipped with an official level editor, and the SDK (software development kit, including source code, tools, and documentation) was released soon after the game. The best total conversions have been showcased annually at the Half-Life MOD expo¹³, and, as already mentioned, some successful/promising mods have been acquired for commercial release. Valve also hosts an official support community for modders, where Valve employees are frequent contributors. Perhaps the most important contribution from Valve, however, was to design a game that is extremely amenable to modding.

Valve's efforts have of course paid off phenomenally. *Half-Life* has spawned what is arguably the most active and persistent creative community in the short history of computer game modding. Almost six years after the original release, despite the aging technology, modding teams are still developing ambitious total conversions for the game. A retail version of *Half-Life* (or one of the commercially released stand-alone mods) is naturally required to play all the numerous free modifications. With the help of the commercial add-on releases, the annual sales of the *Half-Life* franchise actually kept increasing for years after the initial release¹⁴. Of course, much of this success can be contributed solely to the *Counter-Strike* phenomenon. It could be argued that *Counter-Strike* was just a lucky stroke of lightning that had little to do with Valve's supportive efforts. On the other hand there are several other mods for *Half-Life* that, while marginal compared to *Counter-Strike*, are still more popular than any mods for other games.

Understandably Valve is promoting the "mod-friendliness" of the upcoming *Half-Life 2* as one of the game's key assets. Although the exact release date for the game is still to be set, information outlining the technical capabilities of the new Source engine has been available at the official modding

¹³ <http://www.modexpo.com/> (Mod Expo was discontinued in 2002.)

¹⁴ http://money.cnn.com/2003/04/23/commentary/game_over/column_gaming/

community site for months¹⁵. Valve has also announced a high-profile partnership with Softimage, whose 3d-modeling software will be bundled with the game¹⁶. Excited by the new possibilities numerous mod teams have already started work on *Half-Life* 2 total conversions¹⁷.

3.2.1. Organization of the modding scene

Looking at the *Half-Life* modding scene one must keep in mind the age of the game. In the fast-moving world of video gaming a six-year-old title is ancient. Though the game has proven to be exceptionally enduring, the most active years of *Half-Life* modding are already behind, and the scene undoubtedly looks different now than it did in 1999.

Half-Life's old age is quite apparent on the game's official web site¹⁸, where most of the links point to web sites long defunct, and the news section has not been actively updated since 2001. What is not immediately apparent, however, is the developer's outspoken support for the modding community. Only a short list of (mostly outdated) links to fan sites is provided. It is understandable that the promotional site is not full of technical support information, but one might expect to find at least a link to the official modding community site and perhaps some promotional section for the community's output.

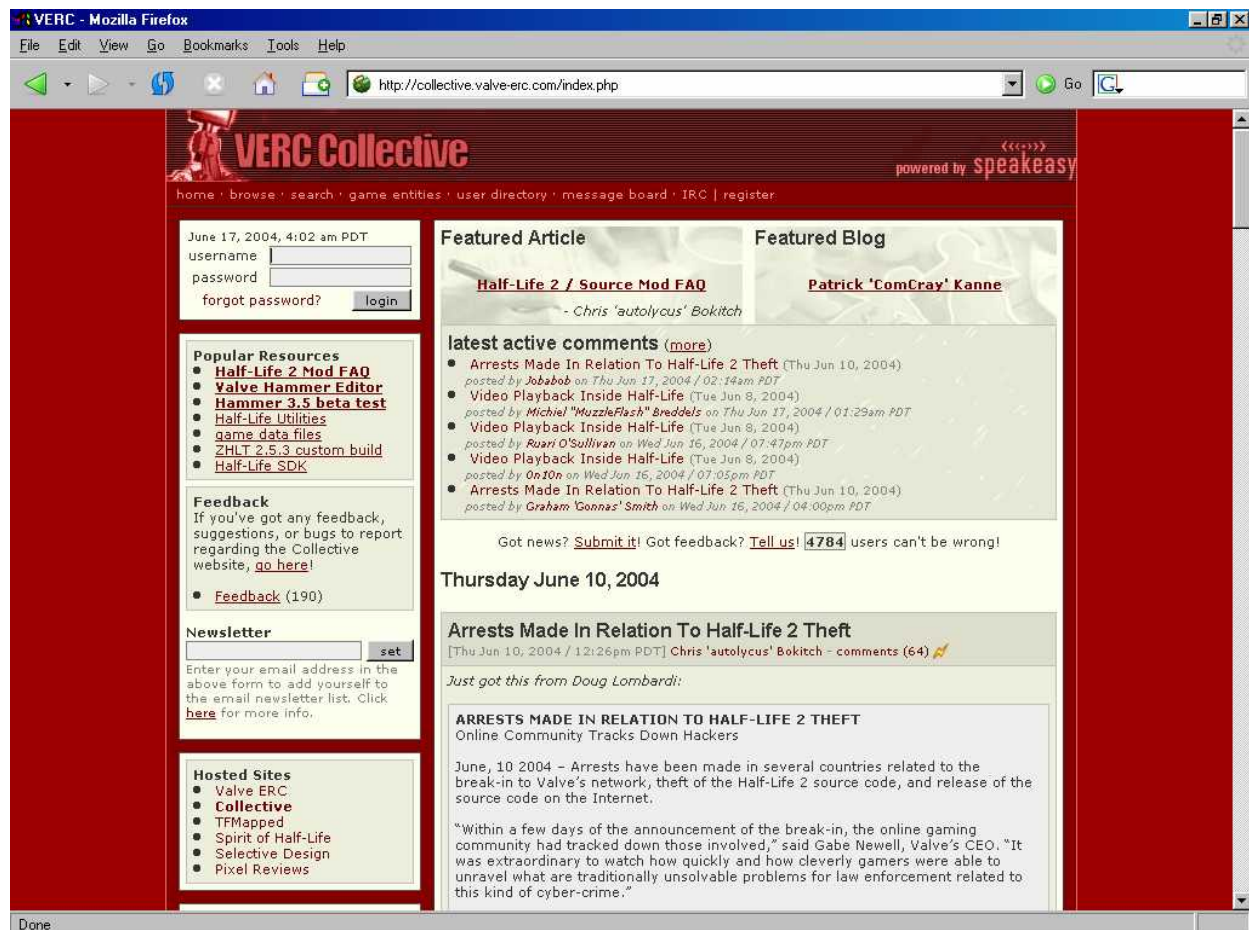
To be fair, the promotional site is maintained by publisher Vivendi Universal, and Valve does offer a link to the official modding community at its own website. As the names suggests, VERC Collective (Valve Editing Resource Center) and its forums are dedicated solely for sharing information related to *Half-Life* editing. There is no (explicitly) promotional material on the site. On the other fan-made mods are also not promoted, save for the occasional posting on the news section. VERC is managed by a Valve community liaison, but much of the content (such as the community news and most of the tutorials) comes from the users. All aspects of modding are covered, though emphasis is on mapping. Most of the information seems to be targeted to modders who already know the basics and want to learn new tricks. Beginners might find it daunting to sort through all the bits and pieces. Of course they can always go directly to the VERC forums for help. Users can also post questions and comments to any news articles and tutorials.

¹⁵ http://collective.valve-erc.com/index.php?go=source_mod_faq

¹⁶ <http://www.softimage.com/products/EXP/HL2/>

¹⁷ <http://www.resistanceandliberation.com>, <http://www.eternal-silence.net/>, <http://www.dystopia-mod.com/>

¹⁸ <http://games.sierra.com/games/Half-Life/>



Current incarnation of Valve's official VERC modding community site came to existence in 2002 when the former official sites Mapping Collective and Coding Collective merged. Impressively, after almost six years of modding, fans are still coming up with new tricks, such as how to play video footage inside the game.

The official VERC site is complemented with several unofficial support sites. The old age of the *Half-Life* modding scene is often apparent on these sites, as many of them are in various stages of hibernation. The ones that are still active tend to focus on some particular area of modding. Some popular sites include Polycount¹⁹, which specializes in 3d modeling and covers many other games also; TWHL²⁰ and Snarkpit²¹, which specialize in mapping; and Wavelength²², which has an emphasis on programming. The content of unofficial support sites usually consists of discussion forums and various fan-produced tutorials. Some also host custom content files.

¹⁹ <http://www.planethalflife.com/polycount/>

²⁰ <http://cariad.co.za/twhl/>

²¹ <http://www.snarkpit.com/>

²² <http://articles.thewavelength.net/>

The screenshot shows the TWHL website interface. The top navigation bar includes links for File, Edit, View, Go, Bookmarks, Tools, and Help. The address bar shows the URL: <http://cariad.co.za/twhl/tutorialbrowse.php?utype=1&subtype=1>. The main content area is titled 'Browse Tutorials' and lists several tutorials under the 'Hammer : Half-Life' category. The tutorials are organized into a table with columns for Title, Description, Author, and Date.

Title	Description	Author	Date
Compiling Introduced	An introduction to compiling, what it is, how it works, and the different ways it can be done.	atom	21/03/2003
Dimensions	How wide, high, long can I make things?	Andy	26/05/2003
How do I run a HL map (.bsp file)	Basic tutorial on using the command line or console to run a map.	Skeeve	10/06/2003
In the Beginning	Really basic tutorial on building a cube and compiling it.	Andy	25/10/2002
In the Beginning Part 2	Walls and Halls: life after the cube. Make another one and connect them with a corridor.	Andy	02/11/2002
In the Beginning Part 3	Basic Texture Application: Got walls and halls, now add some textures to make it all look great.	Andy	06/11/2002
In the Beginning Part 4	Basic Lighting: Got a textured room, so add some lights...	Andy	06/11/2002
In the Beginning Part 5	Furniture: Add some furniture prefabs to enhance realism.	Andy	04/12/2002

TWHL, one of the popular unofficial support communities, has categorized its fan-made tutorials according to skill level requirements. The discussion forum and IRC channel provide help for specific problems and cultivate communality. The site also has a large archive of custom maps, as well as open-source maps for learning purposes.

Besides the community sites, modding support is available on numerous personal web sites, where all the content comes from the same individual modder. Personal sites provide tutorials or tips on the modder's specialty, whether it is mapping, modeling, coding or something else²³. A unique example of a personal contribution to the *Half-Life* modding community is the Handy Vandal's Almanac²⁴. Maintained since 1999, the site includes a thorough encyclopedia of *Half-Life* modding, with all entries (such as Animation, Coding, Level Design) cross-referenced and linked to various external information resources. It should be mentioned that although the content on community sites may come from various people, the sites themselves are also usually started and maintained as a labor of love of one individual modder. If the founding father loses his interest in *Half-Life* modding, or is too busy with real-life issues, the future of the site does not look bright. The sites that already have a strong community around them have a better change of survival

²³ See <http://blaz.csbanana.com/tips.htm>

²⁴ <http://www.karljones.com/half-life/almanac.asp>

in such situation, as the administrator can pass the duties to some community regular(s).

Many aspects of modding – such as 3d modeling and programming – of course involve tools and techniques that are used in various other forms of digital media production. Modelers, for example, can therefore find valuable support from the numerous general 3d modeling communities on the Internet²⁵, as well as from the support forums of popular modeling tools.²⁶

In addition to the support information, *Half-Life* modders naturally also need a forum to share and promote their works. Most user-creations can be distributed quite comfortably via personal web sites because the file sizes are relatively small. More prolific modders sometimes register personal domain names for showcasing all their creations²⁷. These ambitious sites could be seen as resumes for potential employees.

Even if user-creations are shared via personal web sites, a common practice is to post an announcement (with a direct link and perhaps some screenshots) to a modding-related discussion forum. Forums are especially useful for showcasing works-in-progress, as the feedback loop is very direct. The fine-tuning of posted works is often done in a collaborative manner, as the author implements the tweaks suggested by other modders. Updated versions (or links to them) are then posted to the same thread.

²⁵ <http://www.3dbuzz.com/>, <http://www.3dcafe.com/>

²⁶ <http://www.swissquake.ch/chumbalum-soft/forum/>

²⁷ <http://www.fuchtelworld.de/>, <http://www.3d-mike.com/>

Sven Co-op Message Forums - Model Showcase

(Moderated by: Hezus, Panzerfaust, PD-KC)
(Users Browsing this Forum: DarthRod)

Announcement - Please Read
Screenshot Contest By Sniper on 19-06-2004

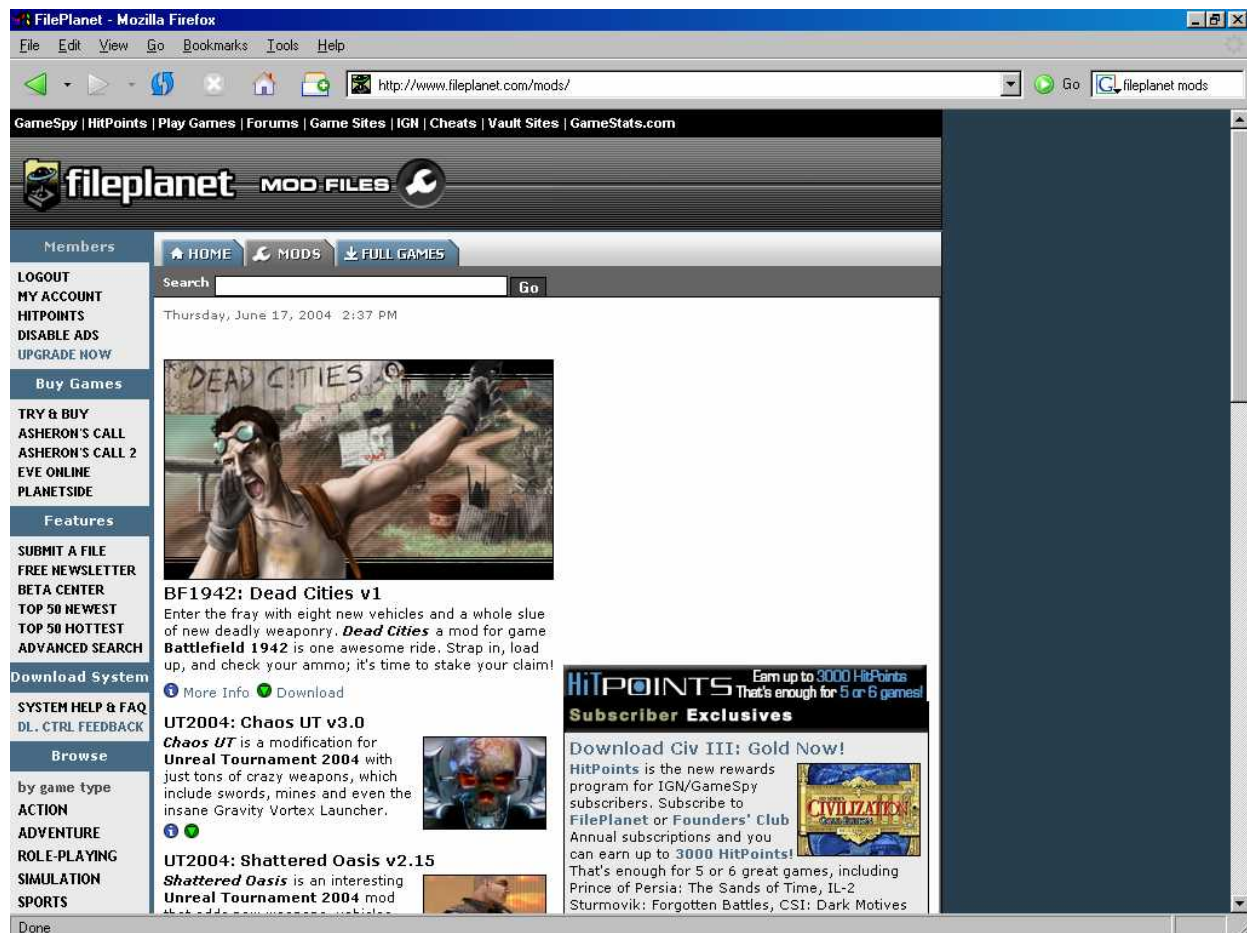
Thread	Thread Starter	Replies	Views	Rating	Last Post
Note: Dead link report thread	Mad Jonesy	0	108		07-04-2004 01:12 PM by Mad Jonesy
Note: Note to all users: Important	Mad Jonesy	0	108		07-04-2004 01:02 PM by Mad Jonesy
Note: Looking for a file? Hints 'n' Tips	Mad Jonesy	0	338		07-04-2004 11:42 AM by Mad Jonesy
Note: Want to post a Model? Read this first	Mad Jonesy	0	160		07-04-2004 03:48 AM by Mad Jonesy
Note: Clarification on Retail model conversions	Mad Jonesy	15	1179		25-02-2004 10:05 AM by Freaking Neo
[Pack]HD weapon v_models (@ 1.2)	Platoon	25	1278		30-06-2004 03:46 AM by Mugsy2002
g[9MMAR] HL w_ and p_ match	Wildcat01	19	193		30-06-2004 03:34 AM by zidane55387
g[Suit]sd_opfor_suit v2	Sgt.Leon	14	341		30-06-2004 01:55 AM by imbettersniper
g[conversion]op4 deagle	Paran0id	11	799		29-06-2004 08:15 PM by Panzerfaust
g[Headcrab] HL2-esque headcrab reskin	Gaming Lord	13	395		29-06-2004 06:43 PM by Agent Cobra
gimproved chainsaw	chubby	3	212		29-06-2004 06:34 PM by DrZero the 2nd

Much of custom content is shared via discussion forums such as the “Model Showcase” on Sven Co-op mod’s web site. Sven Co-op’s forum allows users to post their creations as attachments if the file size is less than 1 MB.

The distribution issue gets a little more complicated when dealing with total conversions. Depending on the amount of custom content (i.e. graphics and audio), total conversion files can be anywhere from 20 to 200 megabytes in size. Few modders can afford the hosting fees for distributing such huge files. Fortunately there are plenty of free gaming file archives available on the web. Archives like Fileplanet²⁸ - the biggest of them all - host user-created content alongside official files such as demos, patches and videos. Besides relieving modders from the burden of bandwidth costs, these archives make it much easier for gamers to find out about mods. A casual gamer might never stumble upon a total conversion fan site, but he is likely to visit Fileplanet occasionally to download the latest patches and demos. There he’ll find new total conversion releases

²⁸ <http://www.fileplanet.com/>. Alternatives include FileFront, FileShack, Jolt, and Gamershell. For added convenience total conversion files are usually uploaded to several archives around the world. Finnish gamers can find many *Half-Life* mods from Peliplaneetta (<http://www.peliplaneetta.net/>).

promoted almost as prominently as official game demos²⁹. The “most downloaded” lists provide further guidance to the world of user-created content.



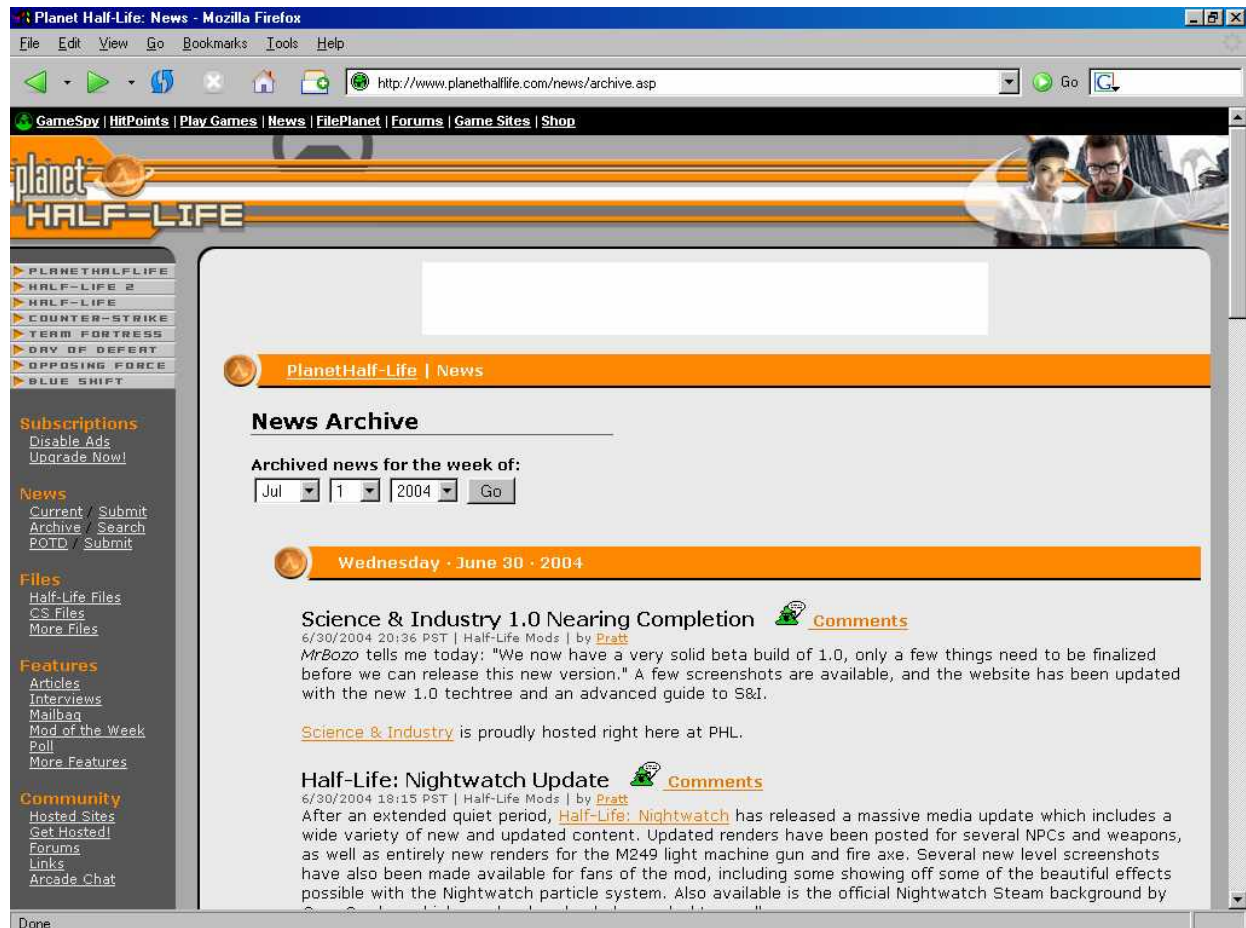
General game file archives such as Fileplanet provide an important connection between mod authors to gamers. A high placing on Fileplanet's “Top 50 Hottest” (downloads) list is perhaps the best PR a mod team could get.

Considering how crucial the success of certain total conversions has been to *Half-Life*'s continued success, it is perhaps surprising that Valve has not set up any official hosting services for *Half-Life* modders. The promotional support for mod teams is also (at least at the moment) almost non-existent. It actually seems that Valve has consciously handed over most fan community services to the Gamespy network. Not only is Gamespy the home of the Fileplanet archive, but it also hosts the most prominent *Half-Life* fan site, Planet *Half-Life*³⁰. PHL is and arguably the most important link between *Half-Life* modders and gamers. Although it is a general fan site and has all

²⁹ Fileplanet hosts mods for dozens of games. Currently the mods for some of the more recent releases have understandably stolen the spotlight from *Half-Life* mods.

³⁰ <http://www.planethalflife.com>. Gamespy Network hosts numerous similar fan sites: Planet Battlefield, Planet Unreal, and Planet Grand Theft Auto among others.

kinds of *Half-Life* content (e.g. gameplay guides), the daily updated news section focuses almost exclusively on the comings and goings of the modding community. PHL also has an expansive review section under the heading “Mod of the Week”. Finally, PHL provides free web hosting services to modders. The hosted sites can be either promotional sites for non-commercial total conversions³¹ or modding-related information resources³².



Planet *Half-Life*, the number one fan site for the *Half-Life* reports daily news from the modding community. PHL is perhaps the most important link between *Half-Life* gamers and modders.

The services Gamespy offers are elemental to the *Half-Life* modding scene. What makes Gamespy a particularly suitable place for such services is the fact that it also hosts some of the world's biggest online gaming servers thereby providing an ideal intermediary between the online gaming community and the modding community (which, as has been already stated, overlap heavily). Obviously, Gamespy does not support modders out of generosity or goodwill, but in the hopes of attracting more paying subscribers for its premium services. (Non-paying users

³¹ <http://www.planethalflife.com/vampire/>

³² <http://www.planethalflife.com/polycount/>

are constricted to slower servers and often have to wait for their turn to download a file.)

Total conversion communities

The community sites discussed so far offer services that are more or less useful to all *Half-Life* modders. However, one of the most distinguishing features of the *Half-Life* modding scene is how it has dispersed along the lines of several high-profile total conversions. Majority of *Half-Life* modders do not actually create content for *Half-Life*, they create it for *Counter-Strike*, *Day of Defeat*, *Natural Selection*, or some other modification³³. Instead of the general community sites, these fans seek support and share their creations on web sites dedicated to the particular total conversion they create content for.

Just like commercial games, well-nigh all total conversion projects have promotional web sites with download links, news about updates, gameplay tips, screenshots, team credits, and discussion forums for support and feedback. The most active areas on the discussion forums tend to be the ones dedicated to “modding the mod”, i.e. the ones where fans meet to discuss content creation for the total conversion in question. These forums are also the place where new user-creations are showcased and announced.³⁴ Members of the mod team usually participate on the discussions frequently. In addition to the promotional sites maintained by the mod team, the most popular total conversions also have several unofficial fan sites, some of which are dedicated solely for modding.³⁵ There are also custom content file archives that are dedicated to sharing user-creations for just the one particular total conversion.³⁶

³³ Among the total conversions, *Counter-Strike* is of course several heads above others. Its overwhelming popularity among modders and players alike makes it tempting to discuss “*Counter-Strike* modding scene” as a separate entity altogether – especially considering its commercial release as a stand-alone game. Nevertheless, since the core engine for all *Half-Life* total conversions, including *Counter-Strike*, is the same, and therefore the tools and techniques used identical, the overall concept of a “*Half-Life* modding scene” is also justified.

³⁴ See for example <http://www.natural-selection.org/forums/>

³⁵ See for example <http://www.dayofdefeat.net/links.html>

³⁶ <http://dodec.hlgaming.com>, <http://specialistscentral.com/>

Natural Selection (Powered by Invision Power Board) - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.natural-selection.org/forums/index.php?s=c7a79cfd111041a318ae396362f331&c=6

**TRANS SYSTEM AUTHORITY
NATURAL SELECTION FORUMS**

Natural-Selection.org Help Search Members Calendar

Welcome Guest ([Log In](#) | [Register](#)) [Resend Validation Email](#)

► [Natural Selection](#) -> [Natural Selection Creation](#)

Welcome back; your last visit was on Jul 1 2004, 07:09 AM

► **Natural Selection Creation**

Forum	Topics	Replies	Last Post Info
Mapping Forum This forum is for mappers interested in making maps for Natural Selection. Ask the creators questions about custom entities and request new features. Give early feedback on other's maps, collaborate on group efforts. If you have Warcraft questions, try the excellent HL-ERC forums instead. Forum Led by: Forum Admins , Forum Moderators , XP-Cagev	5667	76286	Jul 1 2004, 06:58 AM In: Co Mapping Contest By: HanzGrub3r
NS Customization THE location for replacement models, skins, sounds, and sprites. Post, download, discuss, and get help here. Nothing R-rated please. Forum Led by: Forum Admins , Forum Moderators	6090	117872	Jul 1 2004, 07:01 AM In: Natalia Seadijn: A Female M... By: Stavesacre
Artwork Forum Post concept artwork for the aliens, marine or environments. Discuss the art look for the game, get feedback on your map visuals from artists here. Create character models or sketches about any aspect of Natural Selection. Feel free to post any artwork here (no nudity please). Sig tests go here. Forum Led by: Forum Admins , Forum Moderators	2147	36356	Jul 1 2004, 05:42 AM In: The Kill By: Zaggy
Fan-Fiction Forum Post your stories or map backgrounds that take place in the Natural Selection universe. Critique stories or read up to learn more about our game world. Content posted in this forum can NOT be re-posted or used without the author's permission. Forum Led by: Forum Admins , Forum Moderators	642	17506	Jul 1 2004, 05:13 AM In: Ns achio (map) By: Nuub

Quick Log In User Name ***** Go

[The moderating team](#) | [Today's active topics](#) | [Today's top 10 posters](#) | [Overall top 10 posters](#)

Board Statistics

Most active fan-producer communities in the *Half-Life* modding scene meet and collaborate on the discussion forums dedicated to a particular total conversion, such as Natural Selection. On these forums fans discuss editing-related issues, publish (links to) their new creations, request and share reference material (such as photos), propose new custom content concepts, etc.

Although the total conversion communities appear to be relatively autonomous, there is also some “cultural exchange” between them. Because the same tools and skills are applicable, most modders are likely to create content for more than one total conversion during their career in the *Half-Life* modding scene. Some forms of custom content, such as character and weapon models, are also often converted from one total conversion to another (though that practice is not always condoned). Mod teams working on total conversions display universal communality of a kind when they promote each other with “affiliate” banners on their web sites.

Besides the various web sites, *Half-Life* modding scene of course also communicates via other Internet channels. Two Usenet newsgroups are dedicated to *Half-Life* modding discussions, and another one for sharing custom content files³⁷. While these modding groups are nowadays more or less abandoned, the general *Half-Life* group is still relatively active. Valve has set up a couple of official mailing lists, one for

³⁷ alt.games.*Half-Life*.editing, alt.games.*Half-Life*.modifications,
alt.binaries.games.*Half-Life*

mappers and another one for programmers³⁸. There is also an official IRC channel (#valveerc) available for modding-related chatting. Some of the unofficial community sites also have their own IRC channels. Private communication methods, such as email and instant messaging are of course extensively used especially among the members of total conversion teams. Unarguably the most unique communication channel in the *Half-Life* community is the *Half-Life* Radio, broadcasting live five nights a week (at other times recordings are played non-stop). The Internet-only service is operated by semi-professional DJs, who report and discuss *Half-Life* related news (including community news) and interview modders, web site administrators as well as people involved with the development of *Half-Life 2*. Listeners can participate on shows via IRC and voice chat.

3.3. User-created game content

As already stated, a very distinguishing feature in the *Half-Life* modding scene is how it has dispersed into several relatively autonomous fan communities dedicated to different total conversions. In other words most *Half-Life* modders do not actually create custom content for *Half-Life*, but for one of its (multiplayer) mods³⁹. Of course, someone has to build the total conversion first, which involves modifying the original *Half-Life* source code. Generally the process goes something like this: a team of modders creates a total conversion → the total conversion gains popularity among gamers → fans start creating additional content for the total conversion (→ some fan-produced content will be added to the next version of the total conversion). The difference between the mod team and the fan community is of course often quite vague. Apart from the one or two founding fathers, the turnover in the teams tends to be high, and new team members usually come from within the fan community.

Whether it is a team working on a total conversion or an individual fan creating new stuff for his favorite mod, the types of custom content *Half-Life* modders produce are pretty much the same and can be classified into following categories: maps, textures, prefabs, map models, skins, character models, weapon models, sprites, sounds, and code modifications.⁴⁰ Total conversions combine most or all of these individual content

³⁸ <http://list.valvesoftware.com/mailman/listinfo/>

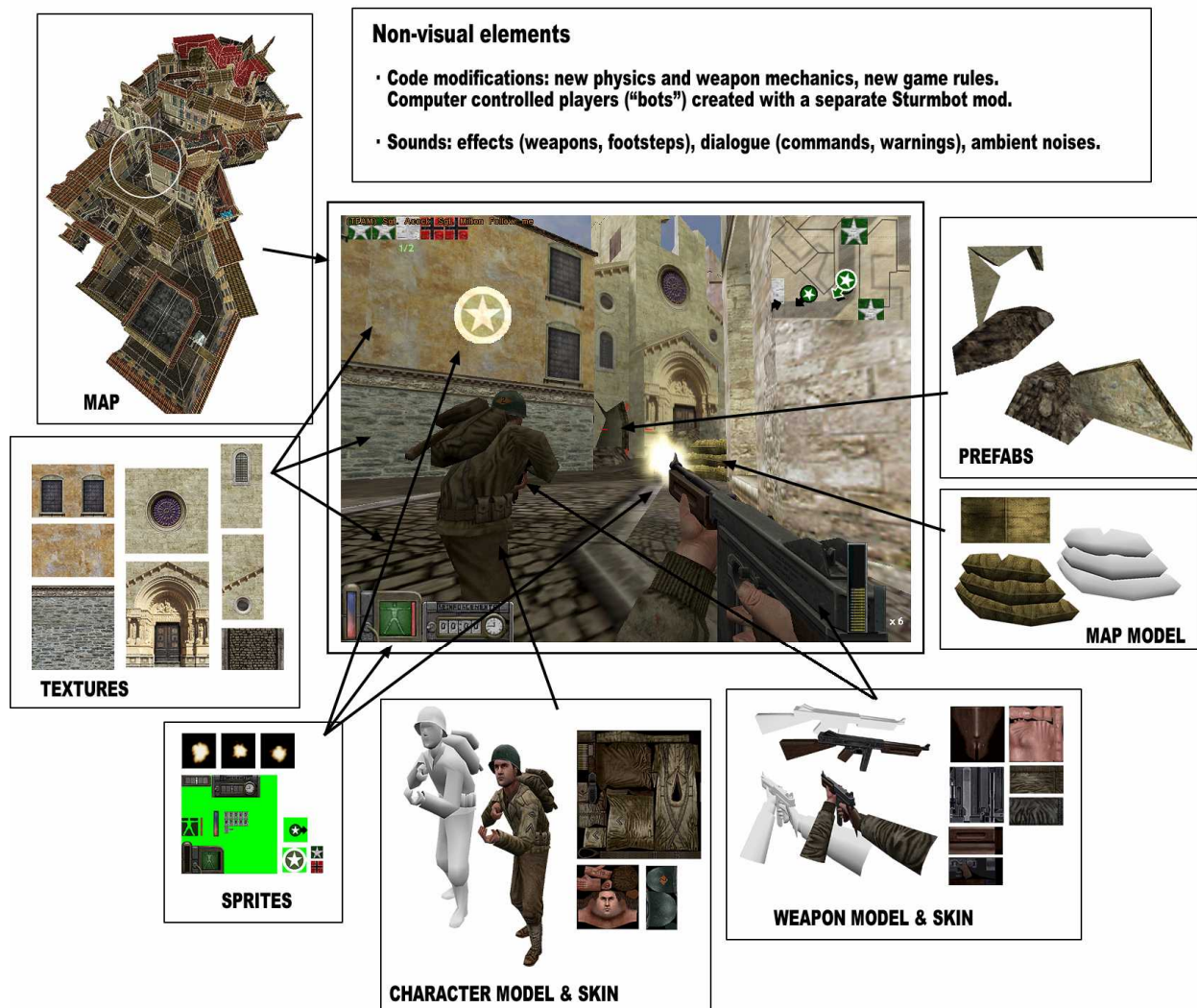
³⁹ Modders do create content for the original *Half-Life* deathmatch also, but this is quite marginal.

⁴⁰ These categories are almost identical in all FPS games and very similar in most games utilizing 3D graphics.

types, while the mod's fan community produces them as separate add-ons or replacements for the original content.

The following dissection of a popular WWII themed total conversion *Day Of Defeat* illustrates how the different forms of user-created content come together.

MOD: DAY OF DEFEAT / MAP: AVALANCHE



Each individual custom content element in *Day of Defeat* serves the same WWII theme. Although the example images are from the official release of the mod, the same thematic restrictions also apply to the *Day of Defeat* fan producers. Modders who want to produce, say, modern weaponry or fantasy characters, have to find other outlets for their creative energy. Fortunately the themes of *Half-Life* mods range widely and every budding game artist should be able to find a suitable canvas. If not, a modder can of course always start building a total conversion of his own with any setting he likes.

The individual custom content types, as well as the tools they are created with, will be discussed in more detail in the following subchapters. Though the example images all come from *Day of Defeat*, after getting acquainted with a few total conversions in the last subchapter, the reader should be able to imagine how the theme of a total conversion might affect the user-creations of each type.

3.3.1. Maps

Maps (a.k.a. levels) are three-dimensional environments where gameplay takes place. In addition to the basic architecture and lighting, a map also assembles all the various usable objects (such as weapons and health packs), dynamic spots (such as player spawn points), and ambient sound sources into place. Single-player maps also need starting points and guide paths for the AI controlled enemies/aides. A custom map may contain any number of custom textures, models, and sounds. These are separate files, distributed in a zip archive with the map.

Maps more than anything define the style and atmosphere of a total conversion mod. The maps in *Day of Defeat* maps for example are inspired by real locations of famous WWII battles. The authors of the official *Day of Defeat* maps have obviously gone to great lengths to create settings that are as authentic as possible. The bar is set pretty high for fan mappers who have to be practically part-time war historians. Total conversions that do not have such an obvious overall theme allow mappers more leeway - inspiration for the map settings can be taken from such sources as popular media (movies, TV shows, and other games) or mapper's own living/working environments. It must be noted, though, that even if the total conversion itself does not have a clear theme, the maps usually do. *Counter-Strike* maps, for example, take place at luxurious mansions, football stadiums, Mediterranean villages, jumbo-jets, and supermarkets. Having a recognizable setting can add a whole new layer to the gaming experience. That said, there are also plenty of experimental "fantasy maps" (especially for *Half-Life* deathmatch), which take place in fictional and/or abstract settings and may also have a strikingly unrealistic visual style.



The thematic boundaries for *Counter-Strike* and *Half-Life* deathmatch maps are loose, which reflects to the diversity of custom maps. Sometimes experimenting with "level artistry" and far-out concepts is even more important than the actual playability of the map. From top left: Morning Star hotel, Medieval castle ruins, Wall-Mart department store, Mid-Eastern palace (style inspired by the PS2 game *Prince Of Persia: Sands of Time* (Ubisoft 2003), flat-shaded experimentation, and a home office setting (scaled so that players feel only few inches tall).

Practically all custom maps created and shared separately (i.e. not as a part of total conversion) are made for *Half-Life* deathmatch or some multiplayer total conversion. Though modding overall is concentrated to multiplayer gaming, there is also a technical reason for this bias. In single-player mods maps have continuity: one map (i.e. level) follows the next in a (more or less) predetermined order as the player advances in the game. Since level transitions are defined in the maps, adding new maps would require that the original maps were distributed

in an editable format, which is rarely the case⁴¹. Each map in a multiplayer total conversion, on the other hand, is self-contained: one battle takes place in one map, and the maps can be played in any order. Online server administrators can choose to host whichever maps they like. If a new custom map is added to the server rotation, it will be automatically downloaded and installed to each player's computer as they join the game. Of course maps can also be downloaded beforehand from the various mapping sites and file archives⁴². Many of these sites review new custom maps for their artistic merits and gameplay value.

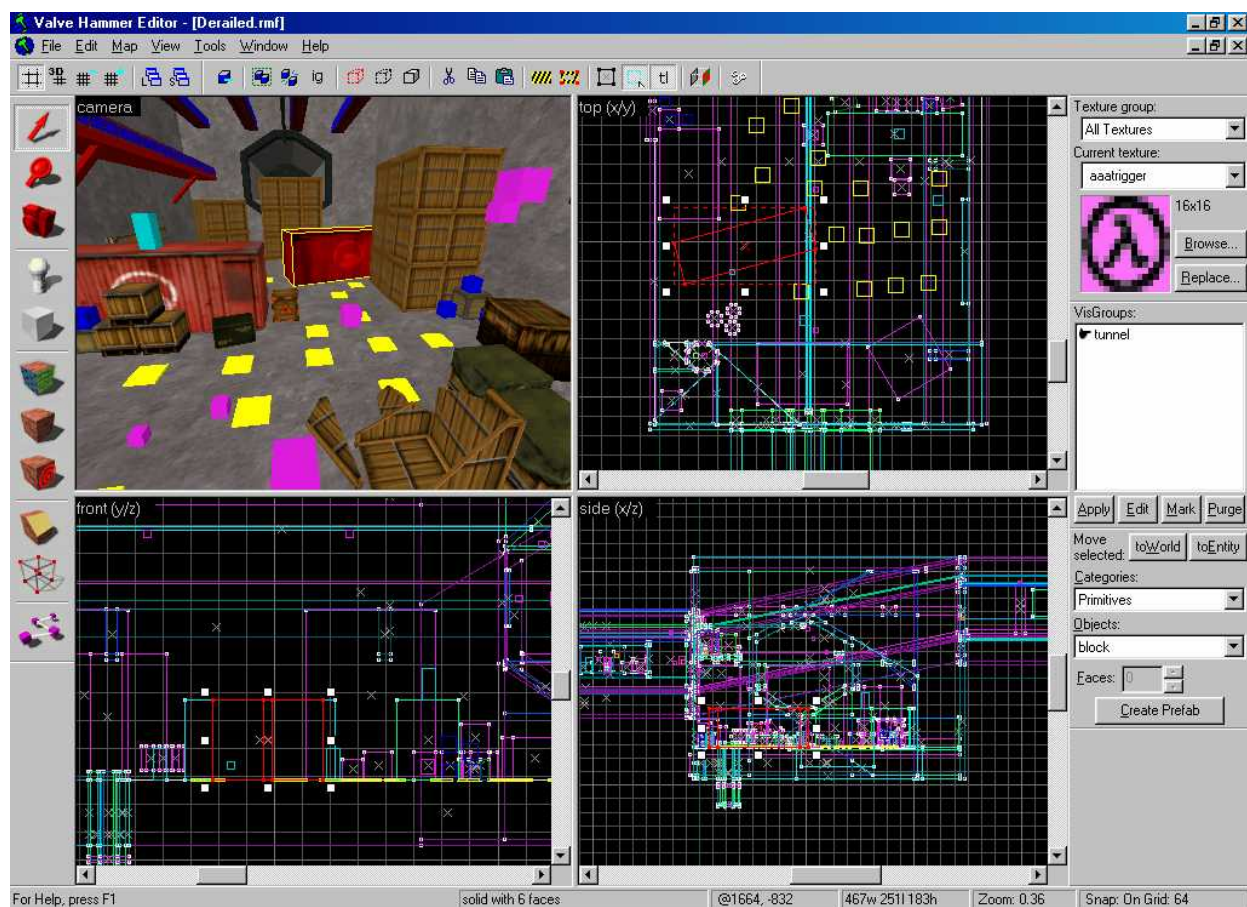
The screenshot shows the CS-MAPS website interface. At the top, there's a banner with the text "ALL YOUR MAPS BELONG TO US". Below this is a navigation menu with links like JOLT, ABOUT, CHAT, BROADBAND, CLANSERVERS, and ÜBERSERVERS. A sidebar on the left contains a "Navigation" section with links to Home, Reviews, Maps, Other Downloads, Skins, Staff, Basic Tutorials, Spray Tutorial, Expert Tutorial, The Game, FORUM, FAQ, Contact Us, and Links. Below this is a "Mirror Sites" section with flags for UK and USA, and an "Advert" for NATIONSTATES. The main content area displays a list of maps with the following columns: Map Name, DL's, Size, Upload On, Site Review, and User Vote. The list includes maps like cs-font, cs-training_maps, csde_bobs_gothic, csde_canali_canel, csde_dust, csde_italy, csde_major-corp, csde_nokia, csde_nokia_nowad, csde_schneiterhau, csde_succionador, csde_unnamed, csdoom, and cstrike. To the right of the map list, there's a "Map Stats" section showing a breakdown of map types: 140 Assassination, 1905 Hostage Rescue, 1677 Defuse (Bomb), 64 Escape, 2020 Other types, 53 Skins, 297 Map Wads, and 1875 Way Points. Below this, it shows 8031 Total Downloads and 15,346,479 Total uploads. The "10 Most Recent" section lists maps like es_losangeles and de_dustXXX.

CS-Maps is perhaps the largest of the many sites dedicated to *Counter-Strike* mapping, hosting over five thousand fan-made maps. Users can also rate and comment the maps, and there are also few hundred editorial reviews.

⁴¹ Custom maps are usually distributed only in compiled format, which is playable, but not editable. (There are de-compiling tools available, but the quality is hit-and-miss, and furthermore, de-compiling someone else's maps is usually not condoned.) If and when maps are released in editable format, it is mainly done for educational purposes, so that other mappers can examine how certain features in a map are created. These "open source" maps tend to be relatively simple and/or unfinished. The official *Half-Life* SDK also includes few example maps.

⁴² <http://www.stats.jolt.co.uk/>

Custom maps for *Half-Life* and its mods are relatively easy to create with the official Valve Hammer Editor (VHE) that comes bundled with the game⁴³. Following the step-by-step tutorial that accompanies VHE, a complete novice can build playable maps in matter of hours. Of course it is one thing to create a map that is playable and quite another to create a map that is actually fun to play. Fortunately the popularity of mapping means that plenty of support is available. Generations of mappers have used versions of the official VHE (previously known as Worldcraft) since the game's release almost six years ago, which means that an immense amount of knowledge has been accumulated to the community. This knowledge is shared on numerous discussion forums and web sites dedicated to mapping.



Valve Hammer Editor is the official map creation tool for *Half-Life*. The CAD-like interface is relatively intuitive to use, though mastering all the features underneath the hood requires plenty of dedication.

Although considered to be one of the better official map editors, VHE has some shortcomings that modders have patched with unofficial helper applications. For example, while Hammer

⁴³ QuArK (Quake Army Knife) is probably the most popular unofficial level editor (<http://dynamic.gamespy.com/~quark/>). Compared to VHE its userbase is marginal.

can compile the map to a playable format by itself, most mappers prefer to use the more efficient and flexible community-produced programs⁴⁴. Another popular utility, Terrain Generator, enables mappers to easily construct outdoor areas for their maps. VHE's geometric primitives (cubes, spheres, etc.) are not especially suitable for creating natural formations.

3.3.2. Textures

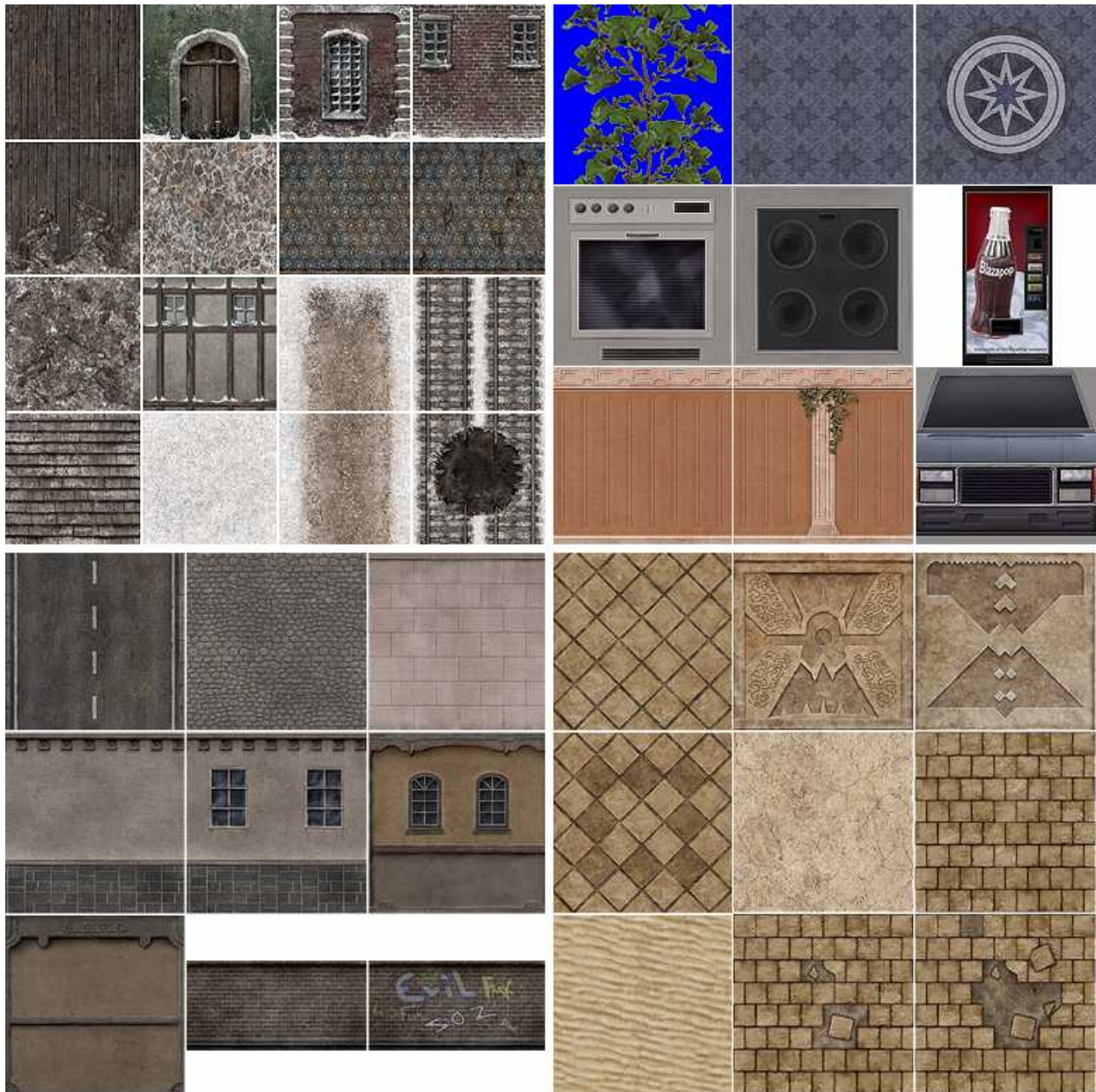
Textures are bitmap images that are applied to the surfaces of the 3d architecture in the map editor. Some textures, such as wall and floor tiles, are designed to cover large surfaces, and form seamless patterns when placed next to one another. Other textures, such as doors and signs, are used as individual pieces. Textures can be animated and partially transparent, as well as emit light.

Novice mappers usually make do with the textures available in the original *Half-Life* or the total conversion they are mapping for. Although this method is likely to produce maps that fit well to the overall theme of the total conversion, it can also easily produce maps that have an over-familiar look to them. This is why more experienced and ambitious mappers prefer to use (at least some) custom textures on their maps. Some mappers have even specialized in texture artistry⁴⁵. Textures can be painted from scratch with an image editor such as Photoshop, but most often they are edited from photographs (like the above example textures from Day of Defeat). Digital cameras have naturally become an indispensable tool for texture artists. Custom textures are often created in sets that follow the same visual theme (e.g. medieval castle). A set might include matching textures for walls, floors, doors, roofs, etc.

⁴⁴ <http://collective.valve-erc.com/index.php?go=mhlt>,

<http://countermap.Counter-Strike.net/Nemesis/index.php?p=2>

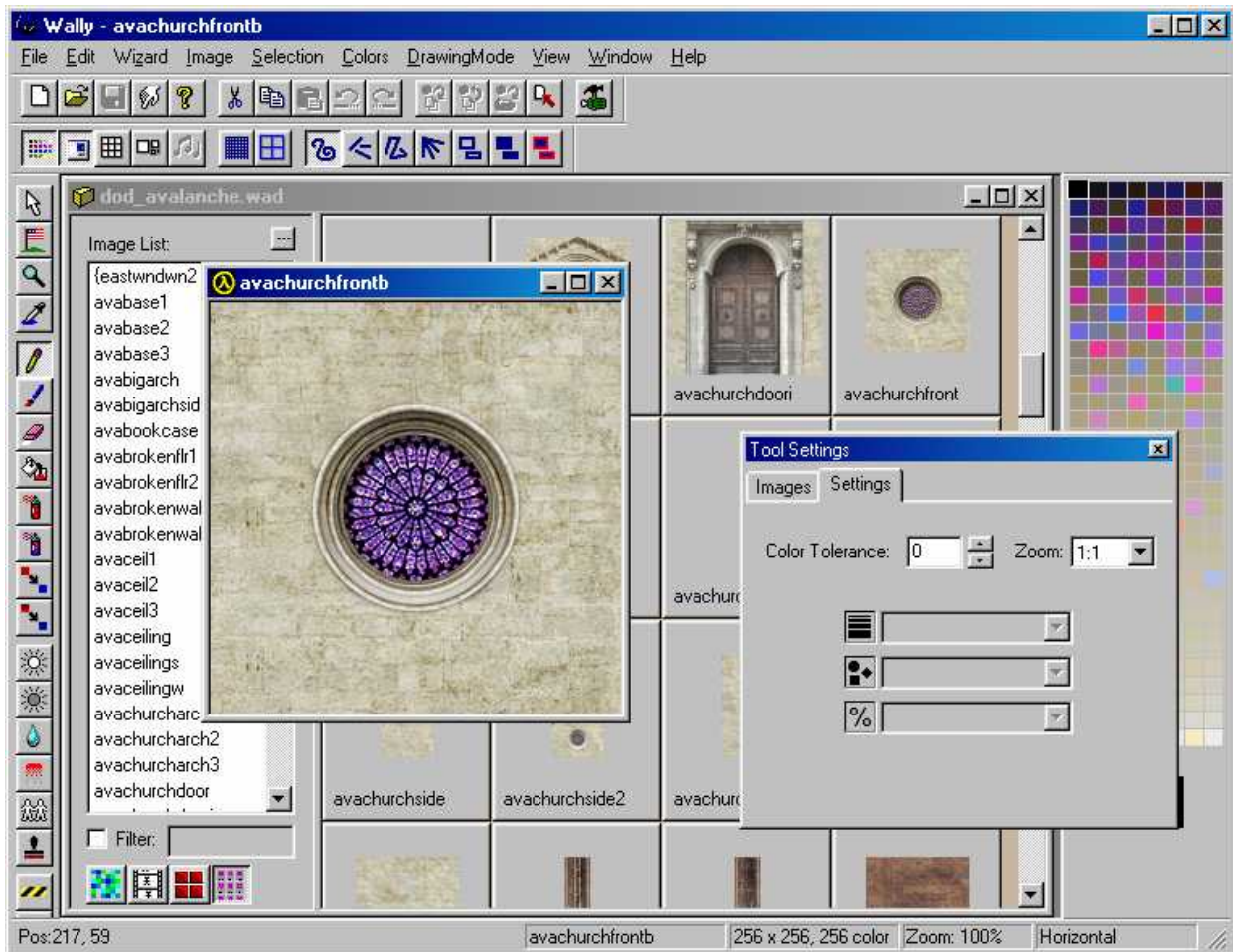
⁴⁵ <http://blaz.csbanana.com/>



Custom textures are usually shared in package files, which include dozens of bitmaps following the same visual/contextual motif. The samples pictured above come from four different high quality sets by renowned texture artists.

While the textures themselves can be created with any graphics application, a specialized utility is needed to compile them into a single WAD file, which is distributed with the map. Most mappers use Wally⁴⁶, an unofficial tool originating from the *Quake* community, which also includes a wide selection of image editing functions.

⁴⁶ <http://www.telefragged.com/wally/>. Like so many *Half-Life* modding tools, Wally originates from the *Quake* community.



Wally is an unofficial utility for packaging the textures into “WAD” files. The contents of downloaded WAD files can of course also be browsed, and texture images exported for editing. Wally itself also has some basic image editing functions.

Mappers who are not capable of creating their own textures but do not want to settle for the existing ones either, can find plenty of custom textures from the Internet. A game graphics archive Wadfater, for example, hosts thousands of textures that can be used freely in non-commercial modding projects. Some textures are also available on the personal web sites of *Half-Life* mappers/texture artists⁴⁷. Since all 3D modeling employs texture mapping, modders can also take advantage of any of the numerous general texture archives of the expansive 3d community⁴⁸. Textures can usually also be very easily converted from other 3d games.

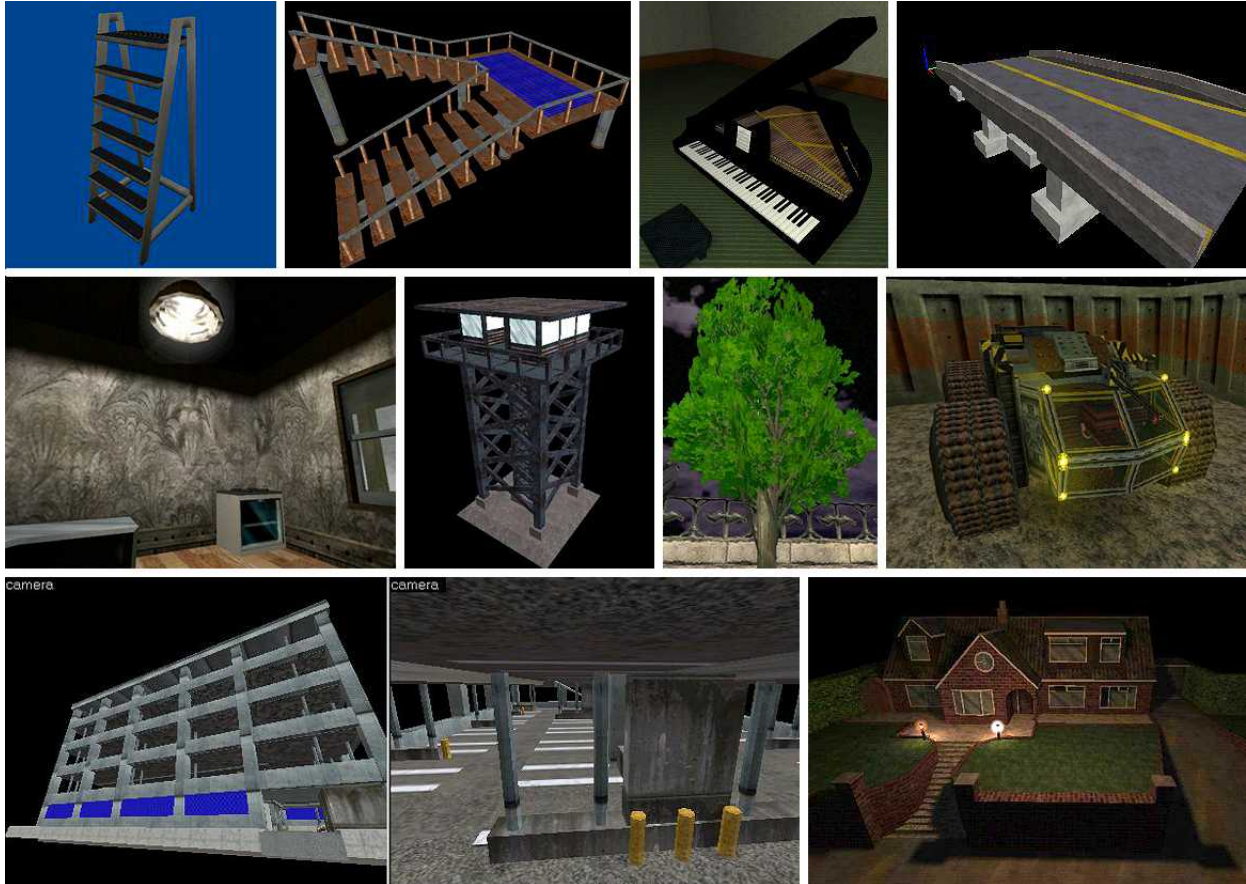
3.3.3. Prefabs

Prefabs are separately saved map objects, which can be easily reused to add various details to maps. Common prefabs include furniture, appliances, doors, pipes, staircases, vehicles, trees, etc. Prefabs are created within the map editor. Since they are

⁴⁷ <http://blaz.csbanana.com/textures.htm>

⁴⁸ <http://textures.forrest.cz/>

composed of same elements ("brushes") as the rest of the map, they can also have similar properties, for example be movable, breakable or illuminating. They can also be interactive and animated (e.g. a sliding door, an elevator).



The complexity of prefabs varies greatly as these examples from Prefabland archive illustrate (from top left): a ladder, a staircase, a piano, a highway bridge, a lighting fixture, an enterable guard tower, a tree, an enterable (but unmovable) vehicle, a parking garage, and a residential house.

The use of prefabs is a somewhat contested issue among *Half-Life* mappers. Some feel that overuse of prefabs is one of the most common mistakes "newbies" make. Prefabs quickly add too much detail to a map, which can cause "lag" on online servers. Another problem is that the same sofas and file cabinets provided with the Hammer editor have already been seen in hundreds of maps and become something of a cliché. Retexturing the prefabs can help, but more ambitious mappers usually prefer to use only prefabs they have crafted themselves for the particular map.

Nonetheless, thousands of ready-made prefabs are available on Prefabland archive site (though many of them are rather poorly constructed). Like textures, prefabs are also sometimes shared in thematic sets (trees, lights, cars, etc.) There have been other prefab archives in the past, but Prefabland seems to be the only one still active. Prefabs are also shared via discussion forums

3.3.4. Map models

Map models are 3d objects that can be used as props in *Half-Life* maps. Unlike prefabs, map models are created with a separate 3D modeling application. This means that map models can be more detailed, but also that they cannot be further edited or re-textured in the VHE map editor.

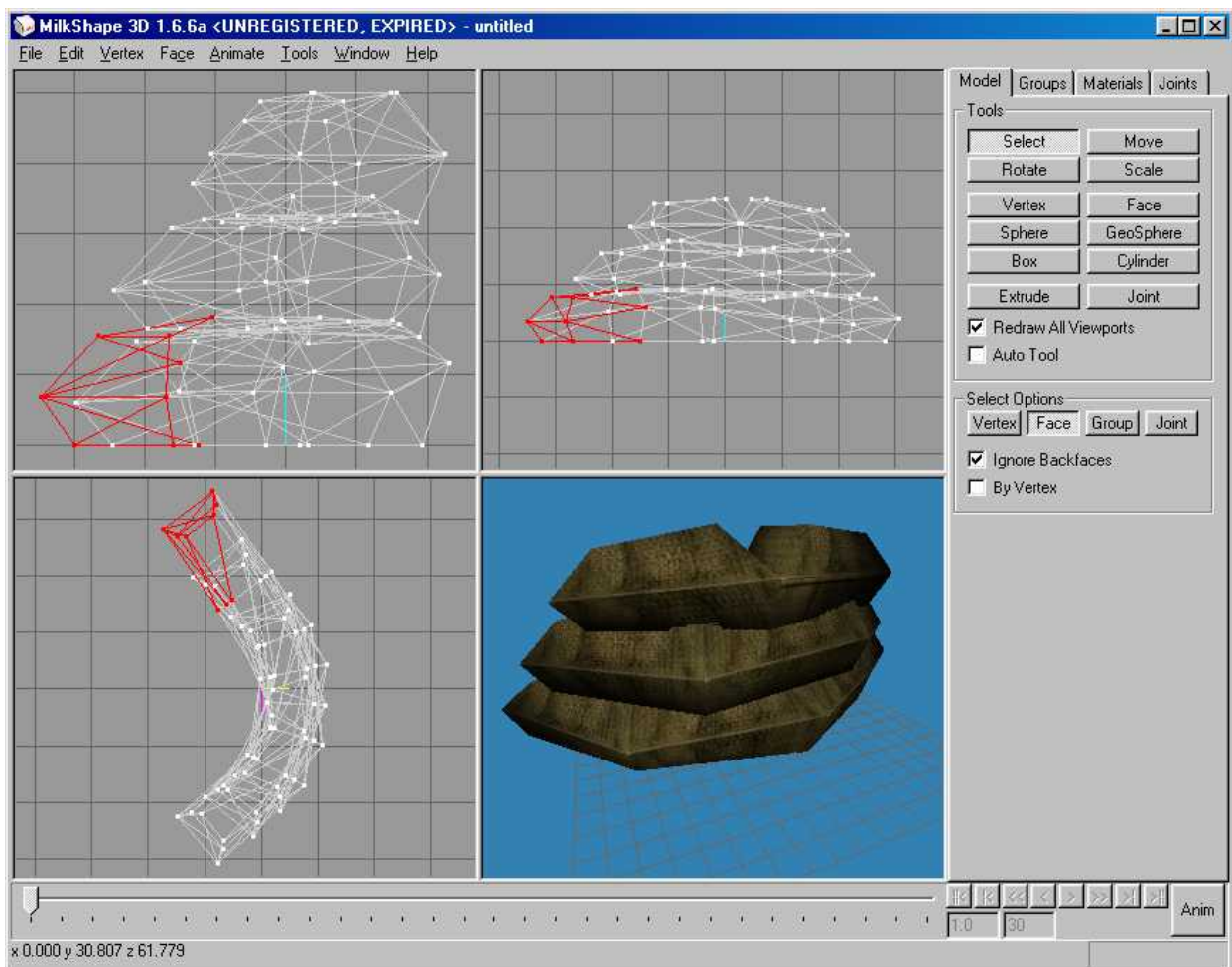
Valve has not provided modders with any official tools for 3D modeling. The SDK did, however, include a utility for converting models created in 3D Studio MAX to *Half-Life*'s file format. Since the SDK has not been updated for a couple of years, the conversion utility does not work with the latest versions of 3D Studio MAX (or 3ds max, as it is currently known). Modders have programmed unofficial exporters themselves, as well as import scripts that open *Half-Life* models to 3ds max.



Map models are used (instead of prefabs) when objects require more detail and/or elaborate animations. The map models pictured above are all for the WWII mod *Day of Defeat*. The wrecked plane and jeep are static models. The pine tree is animated to fall down as if hit by artillery fire. The detailed torn flags have waving animations.

Although 3ds max seems to widely used among *Half-Life* modelers, it is a very expensive tool that most modders cannot

afford (legally). A very popular and much more affordable alternative is MilkShape⁴⁹, developed by a Swiss fan especially for *Half-Life* modeling. MilkShape can be purchased for 25 euros (compared to ca. 5000 euros for 3ds max), and is perfectly capable for the low-polygon modeling and simple animating needed for *Half-Life* models. Furthermore, Milkshape's compile/decompile feature allows modders to import model files from many different games and - with varying amount of tweaking - convert them to the *Half-Life* format. New import/export plug-ins come from the users constantly as new games are released.



Affordable yet feature-rich Milkshape is a very popular modeling tool among *Half-Life* modders. Even modders who use some other tool for modeling and animating usually use Milkshape for decompiling and compiling models to *Half-Life* format.

Very few map models are distributed separately (i.e. not embedded into a map), though some are available on personal web sites⁵⁰. On the other hand any *Half-Life* model can also be used as a prop in a map. Character models, for example, can

⁴⁹ <http://www.swissquake.ch/chumbalum-soft/>

⁵⁰ <http://www.xenondesign.de.vu/>,
http://www.wunderboy.org/3d_games/dod/map_models.php

double as dead bodies. Models found from Internet's general 3d modeling resources can usually also be converted to a *Half-Life* map models with Milkshape, though the level of detail may have to be reduced.

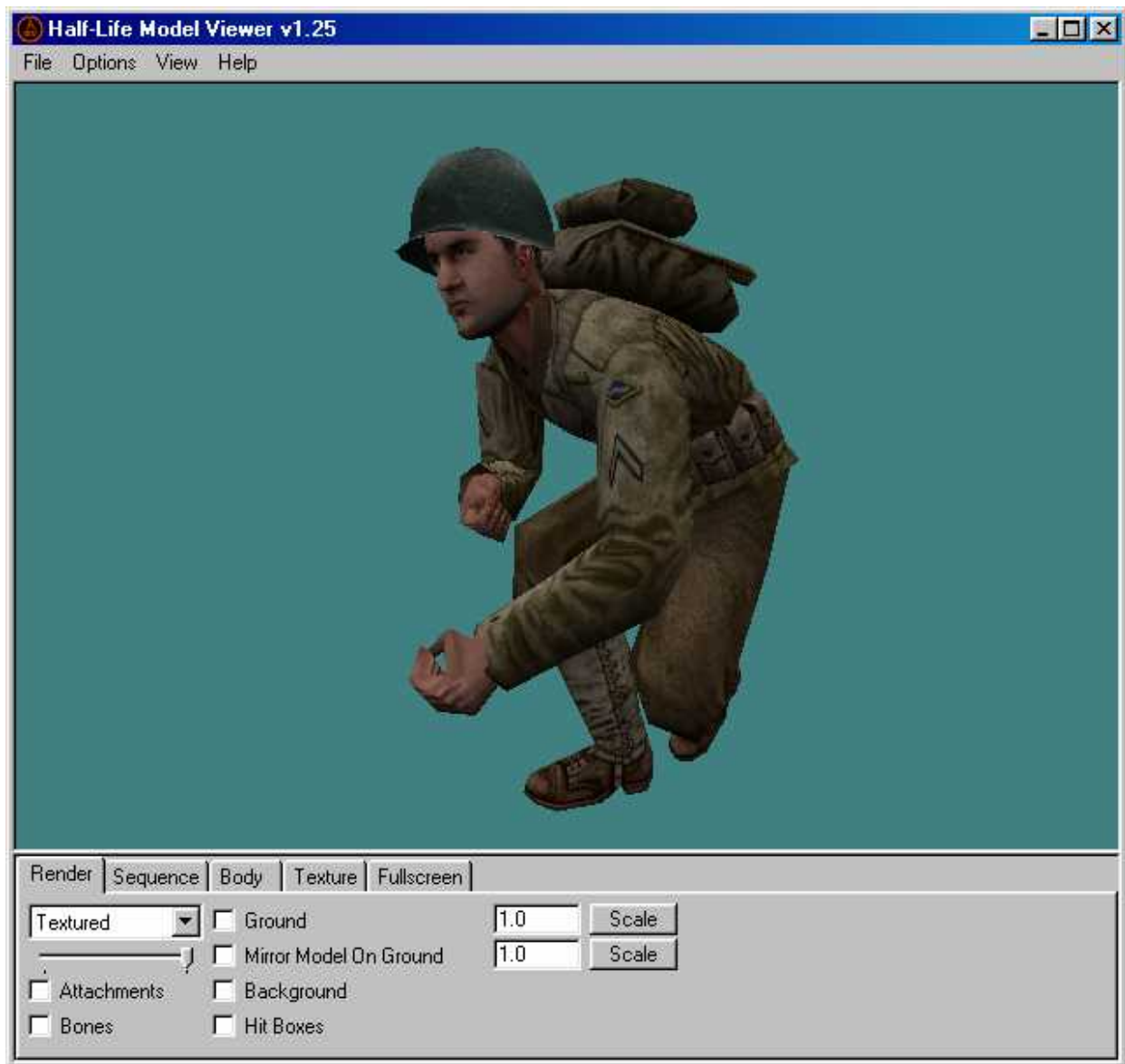
3.3.5. Custom characters (models and skins)

Characters models are animated 3D meshes, and skins are separate 2d images that wrap around the meshes. The model defines character's physique while the skin is like an outfit that the model wears. A single *Half-Life* character model usually has at least two skins: one for the head/face and one for the body. This allows more detail to be applied to the face area. Complicated models can have even more skins. Character models can also have few alternative skin sets; these are needed in teamplay matches to distinguish players from each other. An accompanying script file defines what sort of skins a model uses.



Custom characters often take inspiration from popular fiction and other games. The above custom model/skin combinations are all for The Specialists total conversion (from top left): Captain Chief (from the game Halo), The Twins (from the movie Matrix), gang member (from the game Grand Theft Auto), Nina (converted from *Half-Life* mod Vampire Slayer), Duff Man (from the TV series The Simpsons), and Frost (a character from an anime model set). All of these models also include some custom animations.

Half-Life custom models are distributed in compiled format, which means that both the 3d mesh and the skin(s) are saved in the same file. Compiled models can easily be de-compiled, the 3D meshes tweaked, and the skin images replaced with new ones. The author of Milkshape has created another extremely popular utility for exporting and importing skins called *Half-Life* Model Viewer. (In addition to the skin export/import feature, HLMV allows modders to preview their models without creating a test map and starting the game.) Using the original models from *Half-Life* or one of the total conversions as a basis for new characters is a common practice. Custom characters created by other modders are also often reworked. Permission is usually asked beforehand and the original author properly credited in the “readme” file accompanying the distributed character.



Half-Life Model Viewer allows modders to preview their custom models and animation sequences without loading the models into the game. Skins can also be exported for editing and the edited custom skins imported and saved with the model.

Custom models and skins can be created either for playable characters or AI controlled enemies (“monsters”). Since most modeling is done for multiplayer total conversions that only have playable characters, custom monsters are rare. Of course the playable characters do not always have to be human: in a mod called *Natural Selection* one of the teams consists only of various types of aliens.

Since *Half-Life* is a first-person-shooter, the player usually cannot see his own character (apart from hands)⁵¹. Installing custom player models for a single-player game is therefore pretty pointless. In multiplayer games, however, players can see

⁵¹ Third person view can usually be accessed with a keyboard command. Some total conversions default to 3rd person view.

each other's characters. Custom models and skins can bring fresh new flavor to online matches⁵². Installation is quite simple: the original model file is just replaced with the custom model file⁵³. The hitch is that if other participants want to see the custom character a player uses, they must download and install the models and skins manually to their computers. Otherwise they'll only see the default character the fan-made model/skin is supposed to replace.

As seems to be the case with almost all FPS games⁵⁴, most user-created models and skins for *Half-Life* take their inspiration from popular culture; recreations of characters from movies, TV shows, comic books and other games are particularly popular. It is quite common to convert skins and models from other games to *Half-Life*. Some total conversions naturally require that fan-made characters also submit to a certain theme. While it might be amusing to play as Homer Simpson in a generic *Half-Life* deathmatch, it would not be appropriate in a realistic WWII mod such as *Day of Defeat*. This does not mean that popular culture cannot be used as an inspirational source, however. Fans of DoD have recreated characters from the movie *Saving Private Ryan* (1998) and TV series *Band of Brothers* (HBO 2001).

3.3.6. Weapon models & skins

Weapon models are separately created 3d models that in the game attach to the character model's skeleton structure. Each weapon in a *Half-Life* mod actually requires three separate models: one is shown when the weapon is on the ground, one when another character holds it, and one when the player himself is holding it. The last one is the most detailed and also includes hands. Just like characters, weapon models are wrapped into one or more bitmap skins.

In an action-oriented game like *Half-Life* weapons naturally play an important role. Total conversion projects often start from weapon design and modeling, and the number of new weapons available in the mod is advertised prominently. As an example, while the original *Half-Life* has thirteen weapons, a mod called *Firearms* has thirty. In fact, many total conversion projects seem to exist only to provide certain type of weaponry - realistic, historical, extremely powerful etc. Although certain

⁵² Some online servers do not allow the use of custom models and skins, because they can be used for cheating. A custom model can be used to make the character bigger (i.e. easier to hit) and a bright custom skin to make the character stand out better.

⁵³ When installed separately, custom models/skins always replace an existing model/skin. Mappers can add new AI controlled characters to their custom maps, though this is rare.

⁵⁴ See Sotamaa's (2003) detailed account on *Quake III* models.

mods attract more weapon enthusiasts than others, in general creating custom weapons is extremely popular among modders.



Custom weapons are quite often shared in sets or “weapon packs”. Two such packs for the mod Vampire Slayer are pictured on the left. The three images on the right depict a rather typical collaborative working process in the *Half-Life* communities. The image on the top is a rendering of a custom shotgun model as it was when the modeler posted it to the Natural Selection forums. The images below it are alternative skinned versions posted few days later by two different skin artists.

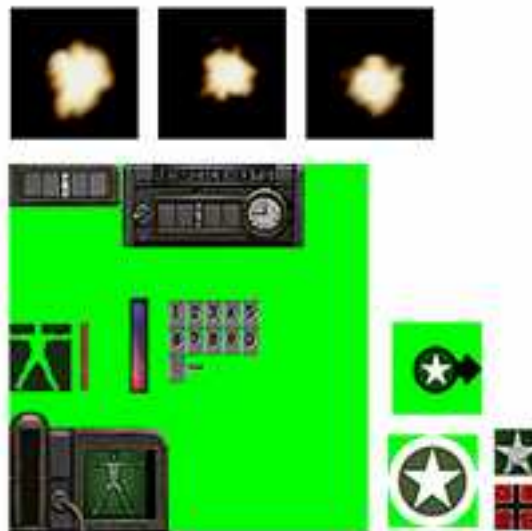
From a technical standpoint custom weapons are similar to custom characters and can be created with same third party modeling and image editing tools. While mechanical objects are generally easier to create than living creatures (animations in particular are much simpler), weapons in FPS games are displayed very prominently on the screen, and any imperfections stand out quickly. Especially crucial in this sense are the skins, because the 3d meshes themselves can only have limited amount of detail. Re-skinning the original models from a total conversion or custom models created by another fan is common practice. Changing the skins can be done with the Half-Life Model Viewer. The weapon creation process (modeling and

skinning alike) is often very collaborative in nature. Reference photos are shared on discussion forums and minute details of the posted work-in-progress screenshots collectively scrutinized.

The theme of a total conversion often sets some limits to weapon modelers - in Day of Defeat the weapons must be from WWII era - but in general the popular objective seems to be to recreate real life weapons as faithfully as possible. Just like character modelers, however, weapon modelers also take inspiration from popular fiction and attempt to recreate weapons seen in various movies and TV shows. Weapons are also often converted from other games.

In addition to the visual models, weapons of course also have physics properties: how powerful they are, how much ammunition they hold, how long are the reload times, how strong is the recoil, etc. These properties are defined in the total conversion's source code and can only be edited by the members of the mod team. Fan-made custom weapons must therefore resemble the original weapons quite closely - a sword cannot be replaced with, say, a chainsaw. Fans can, however, create custom animations for their weapons as the animations are saved along with the compiled model.

3.3.7. Sprites



Sprites are special-purpose 2d graphics/animations that are used for various in-game effects (flames, explosions, laser beams) and interface elements (radars, weapon icons, ammo counters, crosshairs).

Half-Life SDK includes a command line utility for creating sprites, but most modders understandably prefer the more user-friendly unofficial tools. Sprite Wizard, for example, guides the modder step-by-step through the process of converting bitmap images into sprites. The images themselves can be created with any graphics application, though an application capable of

previewing animations is preferable. Modders' favourite image editors Photoshop and Paint Shop Pro both include animating tools.

Custom sprites are generally only distributed as part of a total conversion or a custom map. Most sprites that are available separately on file archives⁵⁵ are static interface elements that change the fonts, icons or the color of the HUD. These sprites can be installed simply by copying over the existing sprite files.

3.3.8. Logos

Logos (a.k.a. decals, sprays) are bitmap images that players can "spray" to walls in multiplayer matches to mark territory⁵⁶. Regular logos are grayscale line-art images, but modders have also figured out a way to use colored logos in the game.

Logos are relatively easy to create. Any generic image editor is enough for simple monochrome logos, but color logos also require a tool like Wally that can convert the images into WAD files. If a gamer is too lazy to create his own logos, thousands of user-created logos can be downloaded from Internet's logo archives⁵⁷. The themes range from popular culture references, to pornographic images, humorous pictures and even political statements.⁵⁸ Some sites have discussion forums where players can request logos for themselves or their clan.

Custom logos seem to be popular among *Counter-Strike* players. One reason for the popularity is that logos are the only type of user-created content individual players can bring into the game that all other participants will automatically see. Player just needs to copy the logo file into his local logo folder and select it for use from the game options. Once the player uses the logo (i.e. sprays it) for the first time, the online server will automatically upload the logo and distribute it to other players. Not all servers support custom logos, and players can also turn off the feature locally.

3.3.9. Sounds

The soundscape of a *Half-Life* total conversion consists of background music, ambient sounds, effects and dialogue. The

⁵⁵ <http://csnation.net/cssprites/>

⁵⁶ Because of their temporality, it is of course debatable whether logos are modifications in the same sense as other forms of user-created content discussed here.

⁵⁷ <http://www.atomicwarrior.com/logos/index.html>,
<http://www.planethalflife.com/counterspray/>

⁵⁸ Wright, Breidenbach & Boria who have studied the diverse forms of communication of *Counter-Strike* players point out an example where a member of the counter-terrorist team used an image of the destroyed World Trade Center as his logo, and in doing so blurred the lines between the game world and the real world. (<http://www.gamestudies.org/0202/wright/>)

placement and volume of ambient sounds is defined in the maps, while effects and dialogue are triggered by various actions players take.

Judging by the forum discussions and file archive contents, sound editing seems to be one of the least popular areas of modding. This might be a false estimate, though, as custom sounds are usually distributed invisibly, embedded into custom maps. Separately shared audio files are mostly weapon effects that attempt to be more realistic/punchy than the original sounds. Sounds are also often taken from movies and other games. Fans of *Day of Defeat*, for example, can replace the original sounds of the mod with music, weapon effects, and dialogue recorded from the TV series *Band of Brothers* (HBO 2001) or converted from the WWII game *Medal of Honor: Allied Assault* (EA 2002). For mods with different themes, movies and games from other genres are poached with seemingly no regard for copyrights⁵⁹. To be fair, sometimes modders put a little more effort to the custom sounds. Some German *Day of Defeat* fans, for example, have recorded a German language replacement for every voice command in the mod. (Obviously it also takes quite a bit of effort to go through a two-hour movie looking for fitting audio snippets and then editing them into usable files.)

Modders can also look for suitable sounds for their projects from the numerous general-purpose sound archives on the Internet⁶⁰ and sound effect CDs.

Replacing sounds is done in a similar manner as replacing models: the custom sound files are copied over the original ones.

3.3.10. Code modifications

The category of code modifications can be divided into two subcategories: programming and scripting. Programming refers to the fundamental changes made into the game rules, physics, mechanics, interface, AI behavior, etc. Programming is done in C++ to the original *Half-Life* source code released as part of the SDK. The dll-files compiled from the modified sources form the technical core of a total conversion, and cannot be further edited by individual fans⁶¹. Scripting refers to the way in which

⁵⁹ The official releases of total conversions usually do not include any copyrighted sounds.

⁶⁰ <http://www.wavcentral.com/>, <http://www.soundamerica.com/>, <http://www.flashkit.com/soundfx/>

⁶¹ Unless of course the modified source code is released to public, which is extremely rare. The only example of an "open source mod" found during this study is the Battlegrounds (<http://www.bgmod.com/>).

mappers make use of the actions defined in the modified program code⁶².

The original *Half-Life* program code can be modified with any basic text editor, though there are also numerous specialized C++ editors available (*Half-Life* modders seem to prefer Microsoft Visual C++), which have features such as searching, auto-formatting and debugging that make the coding process more efficient. In addition to code editors, mod programmers also need compilers that translate the raw code into a working subprogram (dll file). Commercial applications are usually combined editors/compilers, but those with limited budget can also find free standalone C++ compilers.

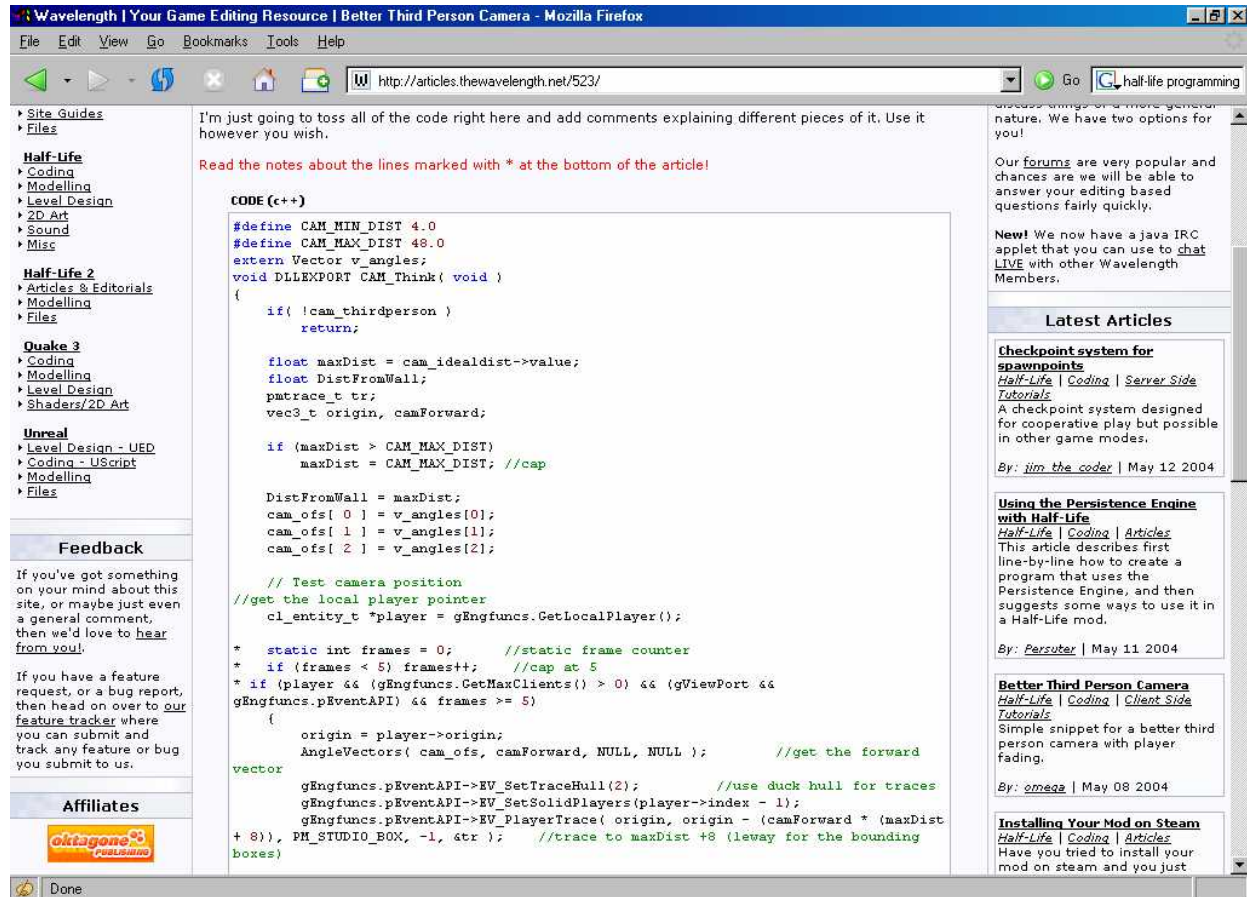
The most important thing that the modifiable subprogram code defines is what kind of game entities (characters, monsters, weapons, supplies, etc.) are available in the mod and what kind of properties/functionality the entities can have (for example, how does an AI controlled character behave when a player enters its vicinity)⁶³. The functions programmed into the source code can be triggered with scripts, which are defined in the map editor. Although the entities themselves cannot be edited once the code is compiled, mappers can play with their properties quite a bit (e.g. which action is triggered, what triggers it, will one action trigger another action, etc.). Most properties can be set from dialog boxes in the Hammer editor, but manual scripting allows more flexibility. Singleplayer mods, for example, often have relatively complex scripted sequences involving AI controlled characters, which can only be scripted manually.

Since the source codes of total conversions are different, some mods allow more diverse scripting than others. A very special case in this regard is a mod called *Spirit of Half-Life*. It's an open source total conversion that extends the original *Half-Life* source with various new entities and functions. Unlike other total conversions, *Spirit of Half-Life* does not offer a new gaming experience, it only provides a new platform to build the mods upon. The mod has been very much a community project and the features added are the ones that *Half-Life* mappers have always wished the original code would have. *Spirit of Half-Life* can be used as a base code for a new total conversion, but custom maps can also be built directly for it. These maps can

⁶² When online gamers talk about scripting they usually mean the practice of binding an often-needed sequence of actions to a keyboard shortcut. *Counter-Strike* players, for example, create scripts for buying weapons, transmitting radio commands and using special movements (such as crouch-jump). Some consider scripting as cheating, others think of it as just a gameplay feature among others. Gameplay scripts can be copied from numerous Internet sites (<http://dynamic4.gamespy.com/~counter-script/>).

⁶³ The VERC site lists all the available entities for the officially released mods: <http://collective.valve-erc.com/index.php?area=entity>.

naturally only be played if the extended *Spirit of Half-Life* version of *Half-Life* is installed. Map scripts are used much more extensively in single-player maps, and most of the new features in *Spirit of Half-Life* are designed with single-player total conversions in mind.



Code modifications are shared on support sites such as Wavelength as example code listings, which can then be copied to the uncompiled source code available in the official SDK. Total conversion teams rarely release their source code to public.

One special type of code modifications are bots, i.e. computer-controlled players that in multiplayer battles stand in for human players. Bots can be used to either fill out the teams if there are not enough human participants present on an online server, or as opponents and teammates in local "single-player" practice matches. Some *Half-Life* total conversions come equipped with official bots, but usually bots come from third party modders that have specialized in bot programming⁶⁴. Some third party bots work with several total conversions, while others are designed specifically for a particular total conversion. Naturally different total conversions set different demands to the "artificial intelligence" of bots. For effective navigation bots always need map-specific waypoint files containing the coordinates of important locations. Waypoint files are often

⁶⁴ <http://www.planethalflife.com/botman/>

distributed independently of bots⁶⁵. *Half-Life* also has a feature that allows players to create their own waypoint files for any map as they are playing it.

Apart from bots, code modifications are not distributed as downloadable files (total conversions are not counted here). Instead, custom code examples are posted to discussion forums and tutorial sites⁶⁶. Players who want to try the custom code must copy the code snippets to the raw source code and compile it themselves. Obviously this is not something casual gamers are prepared to do, so code modifications are mainly published to service other *Half-Life* modders, especially those developing their own total conversions.

3.3.11. Total conversions

Total conversions (referred to simply as mods/modifications in *Half-Life* modder lingo) combine most or all of the above forms of custom content to create entirely new gaming experiences. The most extensive *Half-Life* total conversions are perfectly comparable to commercial games built on a licensed *Quake* engine. And as has already been mentioned, some have indeed been released as standalone games. The totality of total conversions of course varies quite a bit. Some do not venture far from the FPS rules and mechanics of the original *Half-Life*, while others move to a whole another game genre. Some have all new graphics and audio, while others make do mostly with the media resources provided in *Half-Life*. What is common for all total conversions, however, and what separates them from “map/mission packs”, is that they use a modified source code.

Total conversions are by no means a phenomenon exclusive to *Half-Life* modding scene. Many well-known total conversions were already created for *Doom*, and since then almost all popular first-person-shooters have been radically revamped in the hands of skilled fans⁶⁷. Nevertheless, *Half-Life* is commonly regarded to be especially amenable foundation for total conversions. A central reason is that the reprogramming of *Half-Life* is done in common C++ language instead of a proprietary scripting language⁶⁸. A real programming language allows modders almost unlimited possibilities for new game mechanics. And since C++ is commonly used in all fields of software programming, it is very well known and powerful development tools are available. A proof of *Half-Life*'s modifiability is that

⁶⁵ <http://www.planethalflife.com/cswaypoints/files.htm>

⁶⁶ <http://www.planethalflife.com/hlprogramming/>

⁶⁷ See for example <http://ludocraft.oulu.fi/airbuccaneers/>, <http://koti.mbnet.fi/fdfmod/>, <http://www.desertcombat.com/>.

⁶⁸ See “Top Ten Reasons *Half-Life* is Still #1” at <http://archive.gamespy.com/top10/february03/halflife/index3.shtml>.

modders are still starting new total conversion projects almost six years after the game's release.



Downloaded total conversions can be activated from an in-game menu. Mods are also often distributed with installers that create direct desktop shortcuts.

The following examples include both single-player and multiplayer total conversions. As already mentioned, practically all separately created custom content (maps, models, sounds, etc.) that has been discussed above is created for multiplayer mods. Another major difference is that single-player mods are rarely updated after they've been released, while multiplayer mods tend to be continuous projects with new versions coming out years after the original release. Quite often these updates include some content that has come from the fan community.

The selected examples of course only scratch the surface of the whole output of the prolific *Half-Life* community in the past six years. For a more thorough overview, good places to start are Planet *Half-Life*'s Mod of the Week feature, and the TenFour web site, which has catalogued and reviewed almost two hundred single-player mods.⁶⁹

⁶⁹ Planet *Half-Life*'s Mod of the Week feature (<http://www.planethalflife.com/features/motw/>). Ten Four (<http://tenfourmaps.telefragged.com/>).



Counter-Strike



Day Of Defeat



Natural Selection



The Specialists



Sven Co-op



Vampire Slayer



International Online Soccer



Half-Life Rally



They Hunger



Poke646



Heart Of Evil



Point Of View



Earth Special Forces



Battlegrounds



Snow War

Examples: Multiplayer

*Counter-Strike*⁷⁰ pits a team of terrorists against a team of counter-terrorists. Depending on the map that is played, the teams have varying objectives, such as hostage rescue, bomb defusing, and assassination. After each round (usually only few minutes) the players are rewarded with money according to their individual performance and the team's overall performance. The reward money can be used to buy weapons and other equipment at the beginning of the next round. When first released, *Counter-Strike* was a more realistic and tactical alternative for run-and-gun online action games (e.g. shooting while moving is inaccurate, killed players remain sidelined until the next round, teamwork is essential). Following *Counter-Strike*'s success "tactical realism" has been the most prominent trend in online gaming (in mods and commercial titles alike).

*Day of Defeat*⁷¹ is a WWII themed team-play total conversion, where the players are recruited either to the Allied or the Axis forces⁷². In some maps the objective is to control territory, in others to complete certain missions. Most (official) maps are recreations of famous WWII battlefields, and even the fictional ones are inspired by real locations. Uniforms and weapon models are carefully researched to match their real-world counterparts. Overall *Day of Defeat* takes the realism even further than *Counter-Strike* and attempts to be as realistic a war simulation as possible. Players can only carry one heavy weapon (such as a rifle) and a limited amount of ammunition. Weapons have heavy recoil and are very inaccurate if the player is moving. Mindlessly rushing players usually die quickly without knowing what hit them. After being killed, players are sidelined until a wave of "reinforcements" simultaneously re-enters the game.

*Natural Selection*⁷³ is perhaps the most popular unofficial *Half-Life* total conversion at the moment. In the mod alien (Kharra) and marine (Frontiersmen) teams battle for territory in various outer space settings. The most distinctive features in *Natural Selection* are the real-time strategy elements mixed in with

⁷⁰ <http://www.Counter-Strike.net/>

⁷¹ <http://www.dayofdefeat.com/>

⁷² DoD has inspired a fitting online tournament called Battle For Europe (<http://dod.battleforeurope.com/>), where players can "enlist" on either the Axis or Allies army. Battles are fought weekly on different fronts. The winner of the tournament is the army who eventually controls the whole Europe. The fate of France, for example, is decided on battles fought on DoD maps simulating French locations.

⁷³ <http://www.natural-selection.org/>

traditional FPS mechanics. One player in the marine team is the Commander who observes the game map from an overhead view. From this viewpoint he can guide the marine squads in battle, supply resources where needed and coordinate structure building and equipment upgrades. To balance things out, aliens are telepathic: they can always see each other, and when one alien spots a marine, all teammates are alerted. Aliens are also fast, have various special movements (such as climbing along walls and ceilings) and can evolve into stronger species. In addition to the Commander mode, *Natural Selection* has quite complex resource trees reminiscent of RTS games.

*The Specialists*⁷⁴ takes its inspiration from *Matrix* (1999) and other “Hong Kong” style action movies, focusing on over-the-top gunfights and gravity-defying kung fu. Traditional deathmatches get fresh new flavor as players use martial arts, do flips and dives while shooting, walk along walls, and - in a patented Hong Kong movie style - activate slow-motion. In the slow-motion mode players can dodge bullets and even kick the weapons off from other player’s hands and then dive in to reach them. Players of course have a huge arsenal of weapons to choose from, each model painstakingly detailed. Most of the weapons can be upgraded with accessories, such as laser scopes. Other (once) popular examples in the *Half-Life* scene include *Action Half-Life*⁷⁵, *The Existence*⁷⁶, and *The Opera*.⁷⁷

*Sven Co-op*⁷⁸ has a different premise from other multiplayer mods. Instead of battling each other, in *Sven Co-op* players work co-operatively against AI controlled enemies. The installation modifies all the original levels from *Half-Life* to support co-operative play, but the mod also includes numerous brand new levels, as well as many custom monsters and weapons. Although players are on the same side, they get points for each kill, which introduces the competitive element.

*Vampire Slayer*⁷⁹ is yet another teamplay total conversion heavily influenced by popular culture - the TV series *Buffy The Vampire Slayer* (Fox 1997-2003) in particular. In *Vampire Slayer* one of the teams plays are vampires and the other as a vampire slayers (humans). Game mechanics have been programmed to honour the vampire lore. For example, vampires can only be killed with a wooden stake through the heart - slayers’ bullets

⁷⁴ <http://www.specialistsmod.net/>

⁷⁵ <http://ahl.telefragged.com/>

⁷⁶ <http://www.planethalflife.com/features/motw/ex.shtm>

⁷⁷ <http://opera.redeemedsoft.com/>

⁷⁸ <http://www.svencoop.com/>

⁷⁹ <http://www.planethalflife.com/vampire/>

only knock them down temporarily. Vampires can also feed from dead bodies to recover health. Vampires do not have weapons, but they are much quicker than slayers and able to sneak up quietly. The murky, urban-gothic visuals and gloomy soundscape of the maps complement the theme to a quite terrifying effect.

*International Online Soccer (IOS)*⁸⁰ and *Half-Life Rally*⁸¹ are modifications that demonstrate the modifiability of the *Half-Life* engine and truly deserve the epithet “total conversion”. *IOS* uses the *Half-Life*’s first-person-shooter engine as a base for an online soccer game. Weapon mechanics have been turned into soccer mechanics such as passing, shooting, and tackling. The mod also recognizes game field boundaries and awards goal kicks, throw-ins and corners. Even yellow and red cards are handed out. Maps are recreations of different stadiums from around the world and characters wear familiar soccer team uniforms. *Half-Life Rally* converts *Half-Life* into a racing game. Nine current WRC cars have been included with surprisingly well-simulated physics (considering that the only usable vehicle in the original *Half-Life* was a simple carriage on rails). Players can even buy all kinds of upgrade parts (tires, exhaustion, turbo, etc.) to the cars for the money they win. Four different racing modes are available, and the tracks (i.e. maps) are set in different environments and weather conditions. In terms of graphics and physics *IOS* and *Half-Life Rally* of course cannot compete with “real” sports games, but what makes them attractive is that they are free and - most importantly - can be played online. *IOS* fans have formed dozens of teams (= clans) and continuously organize tournaments on the *IOS* discussion forums. Fan base for *Half-Life Rally* is little more marginal, however, and servers running the game can be hard to find.

Examples: Single player

*They Hunger*⁸², released in three parts in 1999-2001, is perhaps still the best-known unofficial single player mod for *Half-Life*. The horror-themed total conversion series takes the player to a small town infested with zombies. All characters, environments and sounds are custom made with remarkable attention to detail. The gloomy visuals, creepy sound effects, surprise zombie attacks and constant shortage of ammo add up to a chillingly atmospheric gaming experience. Although the *They Hunger* series borrows heavily from John Romero’s zombie movies, as well as from survival horror games, such as *Resident Evil* series (Capcom 1996-), its thoroughly professional

⁸⁰ <http://www.planethalflife.com/ios/>

⁸¹ <http://www.hlrally.net/>

⁸² <http://www.planethalflife.com/manke/theyhunger.htm>

presentation has made it an oft-cited example of the creative potential of modding communities.⁸³

*Poke646*⁸⁴ is another single player total conversion with utterly professional production values. The story of *Poke646* takes place some time after the original *Half-Life*. A new hero, Damien Reeves, has to stop another alien invasion, this time in an urban environment instead of a research facility. The atmospheric city maps with their detailed custom textures are of exceptional quality, arguably more impressive than anything seen in the original game (the textures are actually higher resolution). All the other aspects of the mod, from the specifically composed soundtrack to the clever puzzles and new weapons - nail gun, squeezable alien maggot - are likewise very polished work. *Poke646* has a unique background for a mod: a German art student developed it for his university diploma (some work, such as modeling and coding was outsourced). Since his graduation the author has worked on numerous commercial titles, including *Counter-Strike: Condition Zero*.

*Heart of Evil*⁸⁵ may lack the technical polish of *They Hunger* and *Poke646* but it certainly makes it up in tongue-in-cheek absurdity (and the presentation is not that bad either). Set in Vietnam during the war, *Heart of Evil* lifts its storyline almost directly from the movie *Apocalypse Now* (1979). Player's mission is to locate and kill renegade Colonel Kurtz (namesake of Marlon Brando's character in *Apocalypse Now*) while at the same time try to survive the attacks of Viet-Cong, U.S. Marine Corps, and - why not - mutant zombies. Accompanying the player on his quest are the faithful security guard Barney and several new weapons, such as a chainsaw.

*Azure Sheep*⁸⁶, *Point of View*⁸⁷ and *Todesangst*⁸⁸ are just few of the numerous single-player mods that tell new stories in the original *Half-Life* universe. In *Azure Sheep* the hero is Barney, a security guard familiar from the opening scene of the original *Half-Life*. Barney's mission is to rescue a female colleague/girlfriend from the alien infested Black Mesa facility. Once the two meet the AI controlled girlfriend will follow Barney for the rest of the game and must be protected. Somewhere along the way Barney/player will also meet and

⁸³ It should be noted though, that the development of *They Hunger* was sponsored by PC Gamer magazine, which got to release the mods on a cover CDs exclusively before they were made freely available on the Internet.

⁸⁴ <http://www.poke646.com/>

⁸⁵ <http://www.planethalflife.com/heartofevil/>

⁸⁶ <http://halflife.multiplayer.it/azuresheep/>,

⁸⁷ <http://halflife.multiplayer.it/pov/>

⁸⁸ <http://www.peppyfool.com/page5.htm>

interact with Gordon Freeman. In *Point Of View* the player character is an alien named Xonxt, fighting against the human intrusion on his home planet. Instead of using weapons Xonxt shoots energy charges from his claws, and instead of collecting health kits he heals himself by sucking out the life-energy of his prey. *Todesangst* is one of the many unofficial “sequels” to the original *Half-Life* story. Shortly after returning from his heroic adventure of the original game, Gordon Freeman learns (in a scripted sequence) that a jealous co-worker is planning to reopen the portal to the alien dimension. Again it is up to Freeman to save the world. While the first *Todesangst* was more a map pack than a total conversion (although it included some new characters and voice acting), a much more expansive sequel by the same author(s) followed.

The benefits of falling back to the original *Half-Life* universe are obvious: not only does it eliminate the demanding task of creating an interesting world, it also allows modders to take full advantage of all the game’s accompanying resources (models, textures, sounds). Of course it is only natural that the fans enjoy playing with the ideas of “what if” and “what happened next” (especially since the ending of *Half-Life* is decidedly ambiguous). The analogue to (traditional) fan fiction, where fans of TV series and movies fill in the gaps of the official storyline, is quite clear⁸⁹.

3.4. Summary and notes

The story-driven first-person-shooter *Half-Life* was released in 1998 to a huge critical and commercial success. Winning numerous game-of-the-year awards it quickly became the landmark all subsequent FPS games have been measured against. The only aspect somewhat lacking was the multiplayer support. Developer Valve quickly addressed this shortcoming with an official, freely distributed multiplayer add-on *Team Fortress Classic*. Since then Valve has released several commercial add-ons to a mixed response, while the amateur mod *Counter-Strike* has been phenomenally successful. The eagerly awaited proper sequel to *Half-Life* is scheduled for the autumn of 2004.

From the beginning Valve has been an ardent supporter of the modding community. The release of *Half-Life* included an official map editor, and the complete SDK with in-depth documentation, more utilities and editable source code was released few months later. An official modding community site

⁸⁹ For *Half-Life* related fan fiction see <http://www.fanfiction.net/list.php?categoryid=599>.

VERC was founded as well, providing modders with a common meeting place.

Valve's support for the community has paid off tremendously. Six years after its initial release *Half-Life* still outsells many recent competitors. Much of this longevity can be contributed to the modding community, especially to the phenomenal success of *Counter-Strike*, a fan-made total conversion that Valve released commercially as a stand-alone game. Another popular *Half-Life* mod, the WWII themed *Day of Defeat*, was given a retail release as recently as in 2003.

Although the VERC community site provides *Half-Life* modders with an official gathering place, in reality the modding scene is widely dispersed. There are dozens of unofficial support communities, many of which focus on a particular area of modding (mapping, modeling, coding). Most active fan-producer communities are those formed around popular multiplayer total conversions. Fans of a particular total conversion communicate, collaborate and share their creations via the discussion forums on the mod's promotional web site. Some of the most popular mods (*Counter-Strike* and *Day of Defeat* in particular) even have several "third party" fan/modding sites dedicated to them.

The custom content *Half-Life* modders create is shared via personal web sites (links and announcements are usually posted on discussion forums), and various file archives. Some popular mods have dedicated file archives where fans can upload their custom content. General game-file archives such as Fileplanet, which host mods for various games alongside commercial content (demos, patches, trailers, etc.) are especially important, as they provide a natural contact to the more casual gamer. Fileplanet is particularly significant for the *Half-Life* community, not only because it is the most popular of the archives, but also because it is a part of the Gamespy network, which hosts the best-known *Half-Life* fan site Planet *Half-Life*.

Half-Life fans produce maps, textures, prefabs, map models, skins, character models, weapon models, sprites, sounds and code modifications. Total conversions combine most or all of the individual custom content types, with the very core always being a modified *Half-Life* source code. Once the total conversion is released, fans can create and share each content type separately. Some of the content types - namely textures, prefabs, map models and code modifications - are only useful to other modders. The dispersed nature of the *Half-Life* modding scene means that most *Half-Life* modders do not actually create custom content for *Half-Life*, they create it for *Counter-Strike*, *Day of Defeat*, *Natural Selection* or some other mod. Practically all custom content is created for multiplayer mods. The use of custom content is restricted on most online servers, however, as they can be exploited for cheating.

The extent of customization in audiovisuals and mechanics in total conversions varies greatly. Because the source is written in

real programming language (C++), *Half-Life* is extremely modifiable. This is one of the primary reasons why mod teams still choose *Half-Life* as a platform for their ambitious total conversion projects over more recent and technically advanced games. Mods such as the FPS/real-time-strategy hybrid *Natural Selection* and the multiplayer football simulation *International Online Soccer* illustrate the modifiability of the *Half-Life* engine. While multiplayer mods (especially team-play-oriented) currently enjoy a much higher profile, there have also been some extremely popular single-player total conversions - such as the celebrated *They Hunger* trilogy - in the six-year-old history of *Half-Life* modding.

3.4.1. Notes

Half-Life has enjoyed continued commercial success ever since its release over five years ago. This is a rare accomplishment in the fast-moving computer game market. Although much of this success can be accounted to the exceptional popularity of *Counter-Strike*, a credit must also go to the modding community and its prolific output as a whole⁹⁰. A consumer who buys *Half-Life* today buys not only an excellent game, but also an instant access to dozens of other excellent gaming experiences.

Several factors have contributed to *Half-Life*'s exemplary mod line-up. Partly this is due to the commercial success of the game. The millions who bought the game formed a vast breeding ground for creative talent. The popularity of the game has also ensured that there is immediately a huge audience for user-created add-ons. Successful mods such as *Counter-Strike* and *Day of Defeat* have further increased gamers' interest in *Half-Life* mods and no doubt inspired countless gamers to try their own hand at amateur game development.

Of course, not all best-selling games become modder favourites. The game itself needs to be "mod-friendly", i.e. easily and extensively modifiable. As the total conversions described above clearly demonstrate, this has certainly been one of *Half-Life*'s biggest strengths. *Half-Life* was built on id's *Quake* engine, which was purposefully designed to be amenable for modification. Furthermore, the use of *Quake* engine in *Half-Life* has also meant that the game was instantly familiar to numerous *Quake* modders - even most of the same modding tools could be used (Hammer level editor, for example). Perhaps the most important thing is that *Half-Life* modders can take advantage of a full programming language when creating game mechanics. This provides almost unlimited possibilities. All in all, designing and producing a total conversion mod for *Half-Life* is not much different than designing a commercial game for a licensed *Quake* engine.

⁹⁰ See <http://gamespy.com/top10/february03/half-life/index3.shtml>

Finally, credit must be given to Valve for its continued support for the modding community. Distributing the level editor with the game, releasing the SDK, maintaining an official support community site, and promoting total conversion at Mod Expos have all contributed to the popularity and longevity of the *Half-Life* modding scene. This does not mean Valve has done a perfect job, however. In addition to the Hammer level editor, modders would no doubt have been happy to see official tools for other areas of modding also (the command line utilities included in SDK are not widely used). And while the VERC site may serve the technical support needs of modders quite well, a fan site with an emphasis on promoting the works of the community would have helped the interaction between modders and more casual *Half-Life* fans. Lastly, Valve could've perhaps provided some file and web site hosting services for modders mod teams. In all fairness, though, game companies usually do not offer such services themselves.

For a computer game, *Half-Life* has been around for a really long time. To the modding community this is a double-edged sword. On one hand the community has had time to grow, mature and accumulate an immense amount of knowledge. Any problem a modder might encounter has most likely been encountered before - and solved (if solvable). Modding tools have also had time to evolve. Glitches have been ironed out and the workflow optimized. The handicaps that come with *Half-Life*'s seniority are mostly related to the aging technology. For one, the dated visuals may feel uninspiring to modders and gamers alike. The aging engine also sets some very concrete boundaries to the modders imagination. The maximum size of the maps, for example, is quite limited, and vehicles cannot be implemented effectively. These shortcomings have no doubt steered many modders to modern games such as *Unreal Tournament 2004* (Epic 2004) and *Battlefield 1942* (EA 2002), and caused the apparent "depopulation" of the *Half-Life* modding community in the past couple of years.

This depopulation is only temporary, however, as the release of the eagerly awaited sequel is already around the corner. *Half-Life 2*, delayed (allegedly) because of a highly publicized source code theft⁹¹, should come out in the early fall of 2004. Almost every aspect of the release has been designed to support modding to its fullest. In addition to an improved level editor, modders will be treated with free modelling software and lip-sync utilities, courtesy of Valve's partnership with Softimage⁹². Several *Half-Life 2* mod projects are already underway and fan sites dedicated to the sequel currently draw heavy traffic. It remains to be seen what happens to the modding scene of the

⁹¹ See <http://msnbc.msn.com/id/5189819/>

⁹² <http://www.softimage.com/products/exp/HL2/>

original *Half-Life* once the sequel is out. It is unlikely that the original will be abandoned altogether – after all, people are still creating mods for Doom.

From the modders' perspective the most interesting software from Valve might not be *Half-Life 2* at all, but a system called Steam. The already operational system has many features related to online gaming: it finds available servers, prevents cheating, notifies when friends are online, and enables chatting while playing. In addition, Steam works as an online distribution network. Valve's own games are available for purchase via Steam, and the system will automatically download updates for the installed games. What is interesting is that the Steam distribution system will be available to amateur mod teams as well. Once Valve has accepted a total conversion for distribution, each of the millions of gamers using Steam will be able to download it directly from an in-game menu. This will undoubtedly make mods much more attractive to the casual gamer. Furthermore, Valve has suggested that mod authors would be able to charge money for the mods distributed via Steam. As Valve's managing director Gabe Newell predicts, the new publishing model facilitated by Steam could considerably narrow the gap between amateur fan producers and professional game developers⁹³. At them moment only the commercially released mods (*Counter-Strike*, *Day of Defeat*, etc.) are available via Steam. The first fan-made product that made it to Steam was *Codename: Gordon*, a tongue-in-cheek platform game inspired by the storyline and setting of *Half-Life 2*.

⁹³ <http://www.salon.com/tech/feature/2002/04/16/modding/>

4. Case Study: The Sims

4.1. Description of the game

Often described as a virtual dollhouse or a people simulator, *The Sims* (Maxis, 2000) is in fact such a unique game that it is difficult to fit in any pre-existing genre. From an isometric perspective, with a simple point-and-click interface the player-god controls the everyday activities and interactions of a virtual neighborhood inhabited by "sims". The core game play consists of socializing with other sims and performing mundane tasks such as eating, sleeping and going to bathroom, while at the same time pursuing a career in one of the various professions (military, journalism, crime, etc.) available. At any time the player can exit the "live" game mode and start remodeling and furnishing *The Sims'* home with the money earned at the job.

The Sims has no obvious goal - the game is never "completed" - but the open-ended nature of the game allows players to set all kinds of personal objectives. One player might concentrate on making a fortune, another one on socialization, and yet another on house building. A very popular way of playing is to try to make *The Sims* "act" through certain scenarios (e.g. fall in love). These scenarios can then be saved as picture stories with the Family Albums feature (screenshots with annotations). Of course, since it is possible to create dozens of sims to play with, all these objectives can also be pursued in parallel. It is also possible to step back and just observe *The Sims* as the rudimentary AI guides them through their daily routines (often with humorous/disastrous consequences).



The Sims has three modes: Live, Buy and Build. The Live mode (top) is the actual game mode, where player tries to manage the basic needs, skills, relationships and careers of *The Sims*. In Buy mode (bottom left) player can purchase furniture, appliances and other items for *The Sims*' home. In Build mode houses can be remodeled or built from scratch.

Since its release for years ago *The Sims* has become the best-selling PC game of all time. What is remarkable is that more than half of *The Sims*' players are women and girls.⁹⁴ The game's leisurely pace, non-threatening interface, offbeat humor, dollhouse aspects, and the traditionally feminine

⁹⁴ http://www.gamasutra.com/resource_guide/20030916/lewis_01.shtml

subject matter of interpersonal relationships have been key elements to the crossover success. The international sales of the game have been helped by localized versions⁹⁵.

The long-awaited sequel *The Sims 2* is due for release in the fall of 2004. In the meantime Maxis has kept the fans (and game retailers) busy with a total of seven thematic expansion packs: *Livin' Large* (2000), *House Party* (2001), *Hot Date* (2001), *Vacation* (2002), *Unleashed* (2002), *Superstar* (2003), and *Makin' Magic* (2003)⁹⁶. Each expansion pack added a number of new characters, items, careers, interactions and playable environments to the game world.

Although EA/Maxis has been accused of milking *The Sims* franchise a bit too far, some of the expansion packs have extended the game quite substantially. *Hot Date*, for example, added a whole downtown area with restaurants and shopping malls, *Unleashed* added pets, parks and gardening, and *Makin' Magic* all sorts of fantasy elements.

The expansion packs cannot be played without the original game, but the packs themselves can be installed independently from each other. The original game has been bundled with some of the early expansion packs for various special edition releases.

In addition to the PC and Macintosh versions, *The Sims* has been ported for all major consoles. Compared to the computer version, the console version of *The Sims* has a more goal-oriented approach. A massively multiplayer online game *The Sims Online* (2003) is available in the United States, but it has not been as successful as expected. Some contribute the failure of the online version to the fact that it omitted the most attractive feature of the original: the possibility to import user-created custom content into the game.

4.2. Overview of the modding scene

The Sims provides a particularly interesting case for this study, as user-created content is at the very heart of the whole *Sims* gaming experience. As the game's lead designer Will Wright (also known for the *SimCity* series) has stated in an interview: "[What] really draws me to interactive entertainment and the thing that I try to keep focused on is enabling the creativity of the player."⁹⁷ *The Sims* does not only enable player creativity, it demands it.

⁹⁵ *The Sims* themselves speak gibberish and their thoughts are expressed with symbols so localizing the game has been only a matter of translating the interface and the manual.

⁹⁶ The European titles of some of the expansion packs differ from the American ones. *The Sims: Vacation* was the first expansion pack to be translated into Finnish.

⁹⁷ <http://gamestudies.org/0102/pearce/>

The first task player is faced with is creating a sim family to play with. This involves constructing the appearance of each sim from the hundreds of available head/outfit combinations, as well as creating their personality based on several adjustable character traits.

Once *The Sims* are created, the second task is building and/or furnishing a house (or a “lot”, since the yard is also included) for the family. The selection of floors, walls, doors, furniture, appliances, decorations, plants and other objects available is exhausting. Many players spend more time tweaking their houses and yards than actually playing the game.

Finally, players can use *The Sims* for story-telling purposes. Screenshots from the playing sessions can be saved with annotations (e.g. dialogue) to “Family Albums”. Everything players create inside the game - *The Sims*, the lots, and the Family Albums - can be automatically uploaded to the official Sims Exchange community site⁹⁸. Other players can then download them and use in their own game sessions.



Players can either select one of the pre-created families or create a family of their own to play with. The personality and appearance of each sim can be customized extensively.

⁹⁸ <http://thesims.ea.com/us/exchange/>

Of course, the kind of in-game content creation discussed above cannot really be considered modding. After all, the players do not extend or modify the original game; they just rearrange the available content to their liking.

However, Wright & Co.'s pursuit to enable player creative does not end with the in-game tools. *The Sims* was also designed in a way that makes it very easy to import unique user-created content to the game. Maxis released official content creation tools months before the game itself. By the game's launch a large library of custom skins, wallpapers, and floortiles was already available on community sites; by now the amount of fan-produced content far surpasses the official output (even with the constant flood of expansions packs).

After the game's release Maxis continued to release several more official tools. These have been decidedly simple, approachable utilities, indicating Maxis's philosophy that *The Sims* modding is for everyone, not just for the "hardcore" computer hobbyists. Not surprisingly, the more experienced *Sims* fans rarely use the official tools, preferring the more versatile and efficient community-produced utilities instead.

Maxis has not released any technical documentation source code to the community, which obviously has limited the possibilities of *The Sims* modding. There are no high-profile total conversions of *The Sims*, just millions and millions of individual add-on items. Nevertheless, user-created custom content shared on the Internet has undoubtedly been just as important for the lasting popularity of *The Sims* franchise as it has been for *Half-Life*.

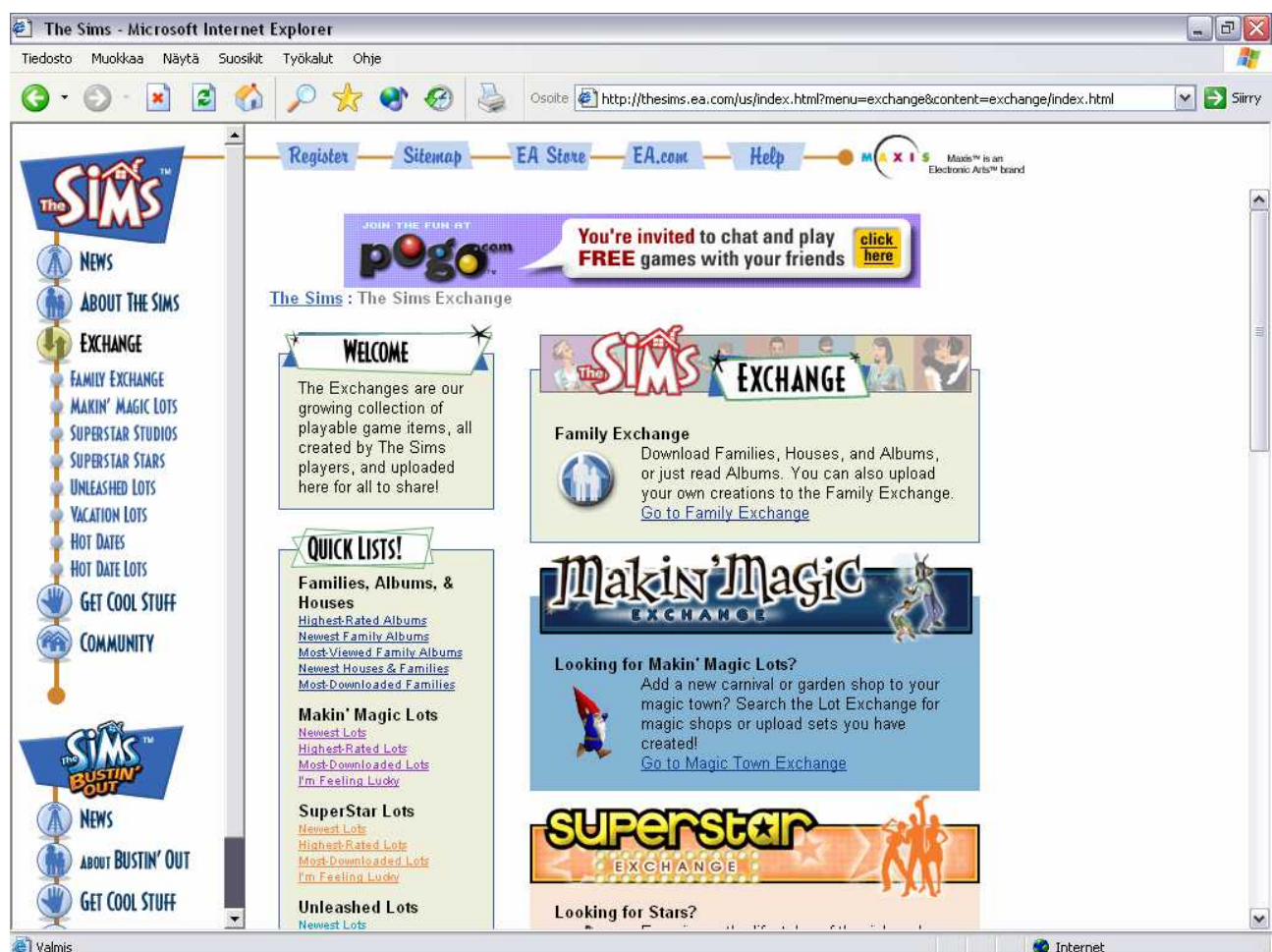
4.2.1. Organization of the modding scene

Like *Half-Life*, *The Sims* is already a relatively old game. In the four years since the game's release the community has had plenty of time to evolve. Unlike the *Half-Life* scene, however, *The Sims* scene does not seem to show any signs of slowing down. On one hand this is probably due to the frequently released expansion packs, which have kept the general interest level towards the franchise high. On the other hand, there is very little competition in the genre that *The Sims* represents. While *Half-Life* modders are naturally tempted by the technologically superior current games, *The Sims* fans have not had any alternative platform for the kind of modding that they practice.

The obvious starting point to *The Sims* fan community is of course the official *The Sims* site hosted by EA/Maxis. The site is directly linked to the game interface: with few mouse clicks players can upload the sim characters, the houses/lots and the Family Albums they have created to the site for other fans to download. The content exchange section is only one part of the

official site. Among all the promotional material (the games can be purchased from the site) fans will find information about the games, official modding tools, some additional game content, an extensive fan site listing, and discussion forums. Live fan chats with Maxis employees are also hosted quite often, especially now that the release of the sequel is approaching.

Surprisingly, there is very little support information for custom content creators. Only custom content installation is briefly covered in the help section. There is not even a dedicated section for modding-related topics on the discussion forums. Since Maxis has not set up a separate modding site - such as Valve's VERC - fan-producers have to look for support elsewhere.



The official Sims site has an Exchange section where fans can share the families, family albums and houses they have created with others.

Fortunately there are plenty of unofficial sites to turn to. The biggest and oldest (established in 1999, before the game's release) unofficial community is The Sims Resource with over 200 000 registered members. For the fans who actively use and produce custom content The Sims Resource (with its sister site The Sims Workshop) is by far the most important meeting place. The site's most prominent feature is the humongous

custom content archive with more than 70 000 files available for download. Full access to the archive requires a paid subscription; this kind of semi-professionalism is very common in *The Sims* modding scene. The Sims Workshop is one of the few community sites dedicated entirely to *The Sims* modding. It hosts a library of content creation tutorials, modding-related discussion forums, and a comprehensive selection of Sims editing tools. Many experienced fan-producers are regulars at The Sims Workshop discussion forums, ready to offer their help to novices.

The Sims Resource - Now over 79400 downloads for The Sims! - Microsoft Internet Explorer

Tiedosto Muokkaa Näytä Suosikit Työkalut Ohje

Ossoit http://www.thesimsresource.com/index.php

Choose a Site: The Sims Resource | TSR Forums | The Sims Workshop | TSW Forums | The Sims Chat

Log in | Register (free) | Create Subscription

ACCOUNT | START | DOWNLOADS | ARTISTS | FIND | TALK | CONTENT | SUBMIT

Frontpage | News | TSR Shop | Competitions | Adventure Game | Calendar | Top 20 | Most Wanted | TSR Wizard Guide

Featured Artist

Xandher | View Profile

One of the most talented and widely known skimmers around, Xandher has probably infiltrated your harddrive with a couple of skins already. From sexy underwear to scary halloween outfits, she...

192 items | 264778 downloads

[View all TSR Artists](#)

Featured news | Jul 22

SecretSims comes to TSR

TSR has made a deal to host all of the fantastic creations from the Secret Sims website, which is due to close. It would have been a terrible shame to see Stina's quality work disappear with her site, so we have acquired it to ensure its life in the Sims 1 community! TSR subscribers now have the added bonus of all of Stina's quality sets, which we will be publishing during the coming weeks.

[View all News](#)

TSR Downloads Archive

Archive	Files	Free	New!	Archive	Files	Free	New!
⊕ Sets	434	13		⊕ Skins	16208	490	
⊕ Animal skins	147	4		⊕ Walls	26494	799	
⊕ Murals	1015	100		⊕ Floors	13240	398	
⊕ Roofs	2533	75		⊕ Objects	9504	288	
⊕ Houses	4508	138		⊕ Downtown lots	413	12	
⊕ Vacation lots	154	4		⊕ Unleashed lots	100	3	
⊕ Superstar lots	90	3		⊕ Heads	941	29	
⊕ Magic Town Lots	21	1		⊕ The Sims 2 TS2!	3679	845	
				In total:	79481	6036	-

Most Wanted Objects (updated every month)

Freebies Archive (the free downloads are rotated every 3rd day from our huge archives)

Top 20 Downloads (updated daily)

SUBSCRIBE TO TSR

- Unlimited Access: Get all 75 000+ files! - Member Profiles: Get your own place

- Unlimited Bandwidth: No waiting in line! - Search Features: Find the files you want

WHY SUBSCRIBE TO TSR?

TSR Installation Wizard

Install/Uninstall anything directly into your game with one click!

Unlimited Access

Download all or any of the 74 000+ files in our archives!

Unlimited Bandwidth

No more waiting in line for downloads during peak hours, you are always first in line!

Special Features

Powerful sort and search features, personal profile

The biggest unofficial fan site The Sims Resource accepts custom content from all fans, but also has a number of regular contributors (see "Featured Artist"). Only a small portion of the file archive can be accessed without paid subscription, but the forums, news and tutorials are free.

While The Sims Resource/Workshop is by far the biggest Sims fan community, there are also plenty of smaller alternatives around the web.⁹⁹ Just like The Sims Resource, almost all of the general fan communities are built around a custom content file

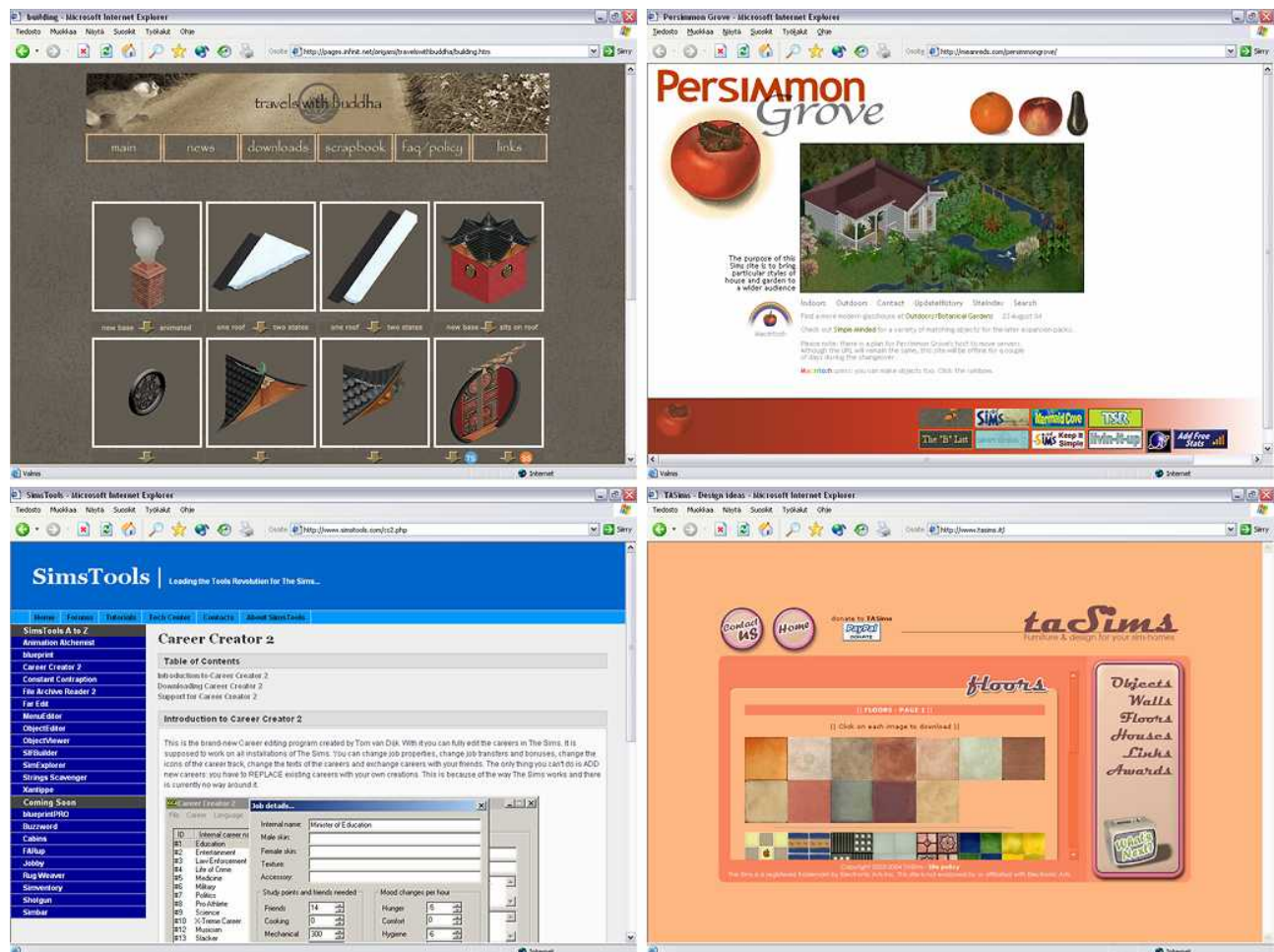
⁹⁹ The official EA/Maxis site has a categorized list of hundreds of sites; other comprehensive listings can be found from The Sims Resource (<http://www.thesimsresource.com/listFansites.sim>), and The Ultimate Sims List (<http://www.ultimatesims.com/>).

archive. The only other feature on most sites is the discussion forum, where (usually) both game play and content creation issues are covered. Some sites also offer few modding tutorials, though none seem to be particularly extensive in this regard. Full access to the file archives is often limited to subscribers (or people who have donated money to the site's maintenance), but discussion forums and tutorials are free. One large fan site worth a special mention is The Sims Zone. Unlike other fan sites, the main feature of The Sims Zone is not to provide file downloads but information. The site reports daily news from official and community-related Sims happenings, and hosts a "knowledge database" where users can post any Sims-related questions. Other users can then post answers in a message forum style.

The fact that *The Sims* fans only produce small add-ons has affected the organization of the modding scene.

First, the lack of total conversions means that there are no distinct total conversion communities like there are in the *Half-Life* scene. All *The Sims* fan sites, as well as all fan creations, are targeted equally to all *The Sims* fans.

The other consequence is that personal sites have a much more central role in the modding scene. Custom content for *The Sims* comes in small files and can be effectively shared via personal sites. Since many fan-producers only share their creations on their own sites, certain personal sites have attracted a very dedicated fan following of their own. Most of the well-known modders have some specialty that they focus on and are famous for in the community. They might, for example, make character skins with a distinct theme (gothic outfits, wedding dresses, celebrities), certain types of furniture (ultra-modern, antique, wooden), "base models" for elaborate outfits and hairdos, or "hacked" objects with customized functionality (these custom content types will all be discussed in more detail below). Of course modders might also be famous for being very talented and for producing exceptionally high quality content. Some fans are also renowned as community toolsmiths. "Team sites", which host the creations of a handful of *The Sims* fans are also very common and often difficult to distinguish from personal sites.



Personal fan sites have a major role in *The Sims* community (from top left): Travels with Buddha offers custom content with an oriental theme. Persimmon Grove has various outdoors objects for sheep farms, vineyards and lush greenhouses. SimsTools is a personal site of a well-known toolmaker. TaSims is a “team site” where two Italian Sims fans share their household designs.

The semi-commercialism of *The Sims* modding scene extends also to the personal fan sites, and quite often some of the downloads are only available to those who have made a donation. Some fans even sell their creations on CDs¹⁰⁰. These fundraising methods are quite understandable, as the hosting costs of popular fan sites can be hundreds or even thousands of euros per month. There is also a specialized web hosting service available for *The Sims* fans. Simshost¹⁰¹ (not in any way affiliated with EA/Maxis) hosts the fan sites for free, but requires a subscription from the users of the sites. One subscription grants access to all The Simshost sites (ca. 60 at the time of writing).

Although there are several sites that specialize in cataloguing *The Sims* fan sites, such listings are really not necessary to navigate the community. Almost all fan sites seem to be a part of some “web ring” and sport link buttons to the affiliate sites prominently on their front pages. Most sites also list dozens

¹⁰⁰ <http://borsaci06.com/cd.htm>

¹⁰¹ <http://www.simshost.com/>

more sites on a separate “Links” page. Another popular promotional practice amongst Sims community is handing out awards to each other. Several sites and forums more or less regularly organize polls where fans can vote for their favorites in different categories. The winners get to put a small graphic on their site as a trophy. It is difficult to find a site that has not won any awards.

Besides dedicated fan sites, *The Sims* custom content is also shared via general file hosting services such as the FilePlanet. This seems to be quite limited however. One reason could be that dedicated fan sites and personal sites allow much more efficient presentation of *The Sims* content.

Finally, it should be noted that Yahoo’s community service Yahoo Groups is particularly popular among *The Sims* fans¹⁰². Yahoo hosts almost two thousand *Sims*-related groups, and the biggest groups have thousands of members. Although the basic function of the groups is to offer a forum for discussions, they also support many other features, such as polls, image galleries, link lists, and file sharing. *The Sims* fans/teams whose web sites are hosted on web servers that do not allow much traffic can use the Yahoo group to distribute their custom content files. Perhaps because it is so easy to set up a Yahoo group, the commitment to maintenance is often weak. It seems that majority of *The Sims* groups have been long abandoned.

4.3. User-created content

As has been mentioned earlier, many of the core game mechanics in *The Sims* were intentionally designed to encourage user creativity. However, the design of the game encourages and supports only certain kind of creativity - namely the production of small-scale add-ons, such as character skins, wallpapers and furniture. Re-programming the game mechanics is impossible, as everything has been hard-coded into the game executable. While it might be difficult to imagine what a total conversion for *The Sims* would look like, there are also some rather surprising omissions in the official support agenda. There are, for example, no official utilities that would enable fans to create character models, animations, objects or object interactions. Maxis has also declined to release any source code or technical documentation that would help the community build such tools for themselves. Nevertheless, resourceful modders have been able to figure out some of the game’s inner-workings on their own. As a consequence, *The Sims* community has been able to produce much more diverse modifications than just the officially

¹⁰² Microsoft’s similar MSN Groups service is also very popular.

supported forms.

The custom content that *The Sims* fans produce falls in the following categories: skins, head and body meshes, walls and floors, objects, hacked objects and houses/lots. Individual custom content items are also often designed in thematic sets; a Victorian living room set, for example, might include period furniture and decorations (objects), wallpapers (walls) and carpets (floors). The most extensive “mod collections” are houses/lots, which can be furnished with all kinds of custom items. The individual items usually can/must be downloaded separately.

The following screenshot illustrates how separate forms of custom content come together in the game. (The house and the farm have been constructed from the custom content items with the in-game tools.)



It should be noted that Sims custom content items often require that a certain expansion pack be installed. Expansion packs have added new objects, character models, animations and

interactions to the game. Many custom content items take advantage of these added features.

4.3.1. Skins

Although the game world in *The Sims* is not three-dimensional - it can only be viewed from four isometric angles - the sim characters are animated 3d models. Each character model is made of two "meshes" - head and body - and two corresponding skins that wrap around the meshes. Additional props, such as hats, bags, and cloaks, can also be attached to the meshes, and these can be skinned separately.

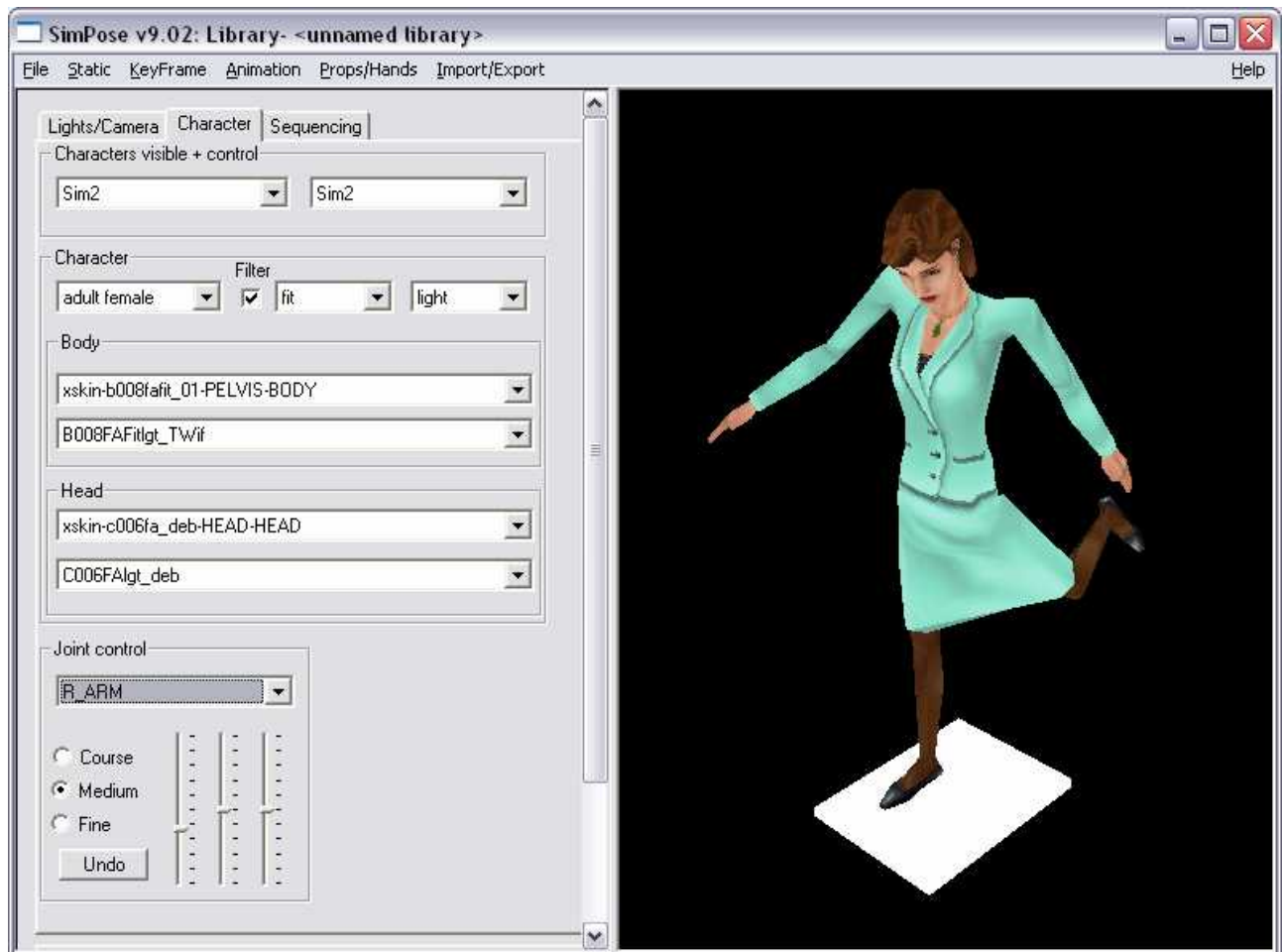
Character skins are the most popular form of user-created Sims content. This is not surprising, considering the games large female demographic. Skinning is also one of the areas where EA/Maxis has been particularly supportive. SimShow, the official tool for previewing and importing custom skins was released well before the game. *The Sims Deluxe Edition* bundle, released in 2002, included The Sims Creator, a versatile skin workshop. Since skins are saved as regular bitmaps, they can of course be created in any graphics application (Photoshop and Paint Shop Pro are popular). Instead of the official SimShow, skimmers generally prefer to use the fan-made SimPose for previewing their creations.



Although the more experienced skin makers prefer to create their skins with image-editing tools such as Photoshop, the official Sims Creator tool has many well-designed features that make skinning instantly accessible – and fun – for the casual fan-producer. With the Face Photo Wizard users can easily create face skins from imported photographs.

The skins Sims fans create could be classified roughly into four categories: fashion skins, thematic skins, fan skins, and faces.

By far the most popular category of fashion skins includes skins that imitate current fashion styles – either trendy street fashions or high fashions seen in celebrity photographs. Skin sites themselves often resemble fashion catalogues, with separate sections for men, women, girls and boys, and further subcategories such as casual, formal, sports, lingerie, swimsuits, etc. The images for these “catalogues” can be created with the fan-made tool called SimPose, which allows modders to pose their sims in any imaginable way. (Pose configuration files are also shared separately on some sites.) While many of the fashion skins are recreations of real world outfits, *The Sims* skinning also provides a popular platform (and a massive audience) for aspiring fashion designers to show off their own designs.



SimPose is a fan-made utility for previewing skins and meshes, posing *The Sims* for screenshots, and creating custom animation sequences. Poses can be saved and shared with other users.

As the name suggests, thematic skins follow a distinct theme. The themes can be as varied as wedding dresses, erotic lingerie, gothic-punk outfits, 60s fashions, fantasy, Victorian costumes, or traditional Oriental clothing. Fan skins are a particular type of thematic skins that recreate familiar characters from popular fiction such as TV-series, movies, comic books, or video games. Some *The Sims* fan sites have specialized entirely on providing skins with a certain theme.

Face skins are skins that only cover the head mesh, giving the sim new facial features. While most of the face skins shared on the net are celebrity faces, there are also plenty that do not attempt to imitate any known person. These can illustrate, for example, different make-up styles, hairdos or ethnic features, but they can also be “ideal” men and women, or available just for the sake of variety. Face skins often require a custom head mesh to be installed also. The mesh is usually created by another *The Sims* modder and has to be downloaded separately. Sometimes the skin may be free but the head mesh is only available at a pay site.

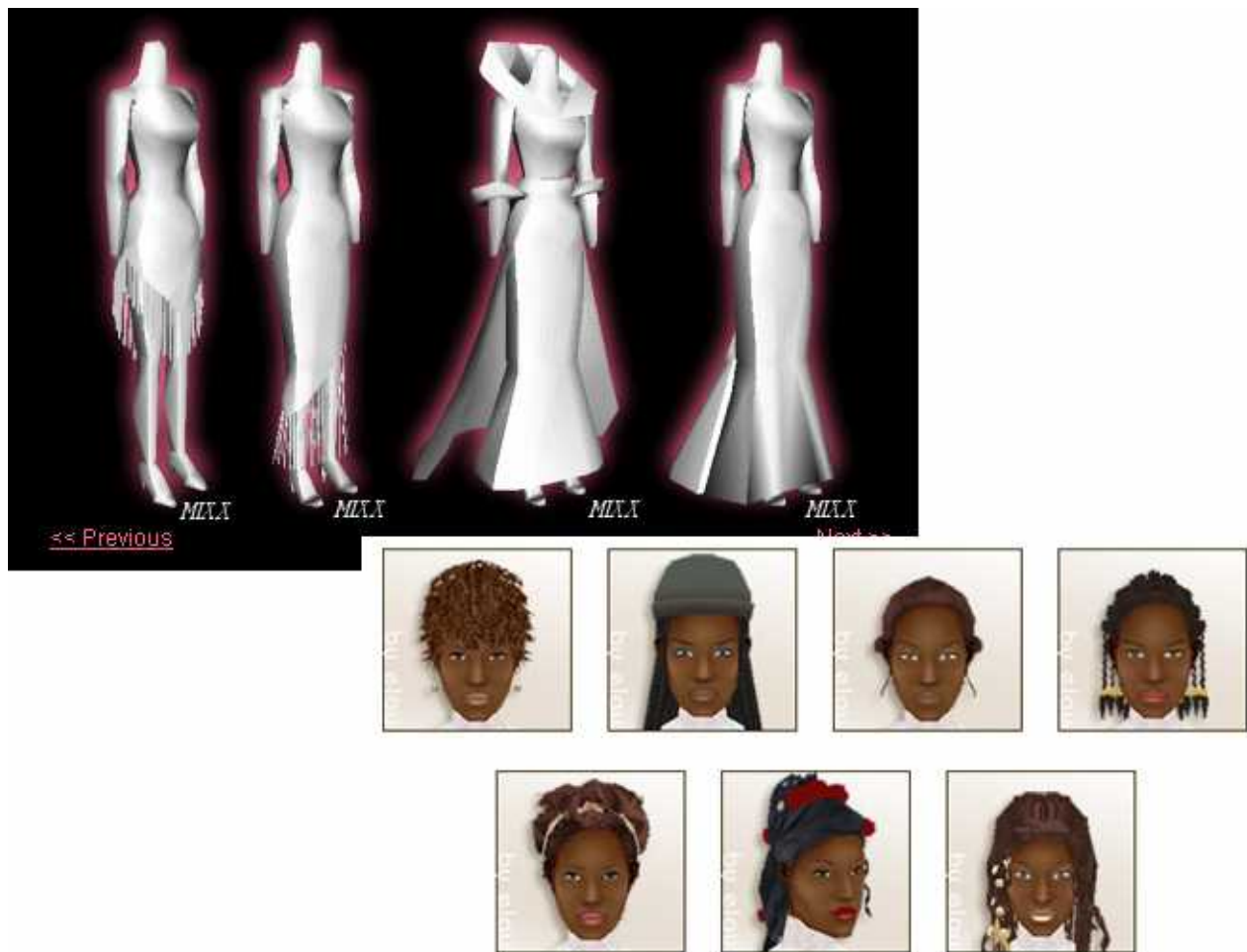


The diversity of custom Sims skins is truly impressive. From top left: Versace Fall 2004 fashions, casual men's wear, traditional Chinese costumes, comic hero skins, Lord of The Rings skin set, porn star celebrity skins, and celebrity faces. All the skins pictured here require custom meshes to work properly.

Installing custom skins is generally very easy. The files are just copied into the Skins folder after which they can be selected in the character creation screen. However, the implementation of skins changed somewhat when the *Hot Date* expansion pack allowed sims to buy clothing. Making regular skins buyable can be confusing and require meticulous file renaming. Perhaps the biggest problem, though, is managing all the installed skins. Self-proclaimed "download junkies" can have hundreds or even thousands of skins stored on their computer. In *The Sims* game skins can only be browsed one by one. Maxis has not provided any external managing tools either, but fans have created few of their own utilities.

4.3.2. Meshes

The underlying physique of *The Sims* is created with 3d models, or “meshes”. Each character is a combination of a body and a head mesh. Optional props/accessories (such as hats, bags, jewelry, cloaks, etc.) can be attached to either of the main meshes. The body mesh dictates the general physique as well as the outlines of an outfit. Different body meshes are needed for different body types, but also for gowns and skirts, pants and shorts, long and short sleeved shirts etc. The shape of sim’s hairdo is sculpted into the head mesh, so different hairstyles require separate head meshes.



“Naked” body meshes from Jeannie’s Bottle of Sims and head meshes with custom skins from SIMPLY Elay.

Since the head and body meshes are separate files, they can be mixed and matched freely. While *The Sims* includes dozens of original body and head mesh types that can be combined in hundreds of different ways, the assortment is very conservative. There are, for example, no obese body meshes or head meshes with spiky hair. The support for custom meshes is not something that Maxis originally had planned, and modders are faced with some unfortunate limitations. Completely new meshes cannot be created at all, and while the original meshes

can be edited, their proportions cannot be altered (due to the way character animations are handled).

Maxis has released one official tool for mesh editing. FaceLift (and later version FaceLift Gold) allows modders to tweak the head meshes. Facial features such as the size and shape of chin, eyes and nose can be adjusted, and the customized head meshes saved into the game. Though entertaining and user-friendly, FaceLift is not capable of exact editing and is obviously limited to the head meshes only. For more precise and substantial head mesh editing, and for editing the body meshes, fan-made tools are required. There are various utilities for extracting the original models from the game data files and converting them into a format recognized by common 3d applications (and vice versa). Milkshape, the affordable low-polygon modeling utility is particularly popular among Sims modders, because of its built-in support for *The Sims* model file format.



With Maxis's official FaceLift tool Sims fans can tweak the existing head meshes in a user-friendly manner. Serious modders use real modeling tools however.

Props/accessories can be modeled and converted with the same applications as body and head meshes. The way in which the

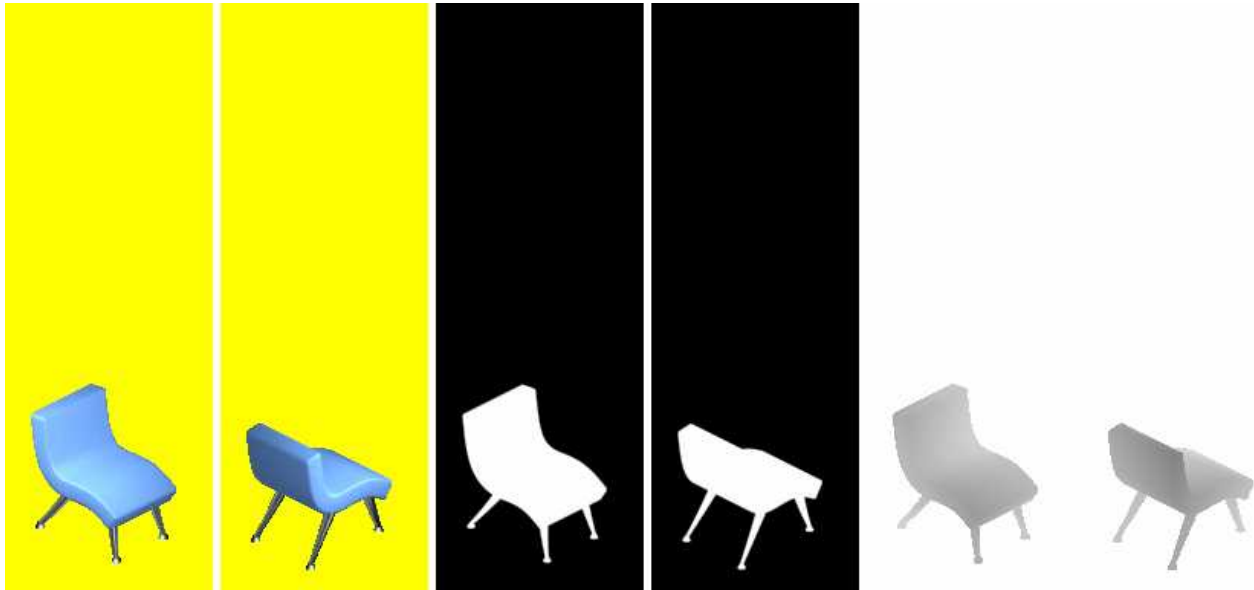
props attach to the main meshes is defined in a text file. Once extracted, the text file can be edited with any text editor, but there is also a specialized tool for the task.

Compared to skimmers, modelers are a relatively tiny group of simmers. Most skimmers do not create their own meshes but take advantage of those circulating in the community. Mesh makers are happy to share their creations, as long as proper credit given. If a skin requires a custom mesh, it is usually noted next to the download link. Sometimes the mesh is bundled with the skin in a zip file, but often the skins have to be downloaded from entirely different web site. The meshes might also only be available for paid subscribers or donors while the skins are free. Any skin can be applied to any mesh but obviously the skins will not fit the “wrong” meshes equally well.

4.3.3. Objects

Objects can be anything from building supplies (walls, windows, fences, etc.) to garden items (plants, trees) to furniture, and electronic appliances. Unlike characters, objects in *The Sims* are not three-dimensional. They are composed of bitmap images taken from four fixed angles. Parts of the bitmaps are made transparent with “alpha masks” to define the objects outline. The “third dimension” of the objects - sims can walk in front, behind or through objects - is achieved with special bitmaps called “Z-buffer masks”¹⁰³. Objects or parts of objects can also be composed of image sequences (an opening door of a fridge for example). With the exception of building components, most objects in *The Sims* have some functionality or “behaviors” attached to them. These will be discussed below under “Hacked objects”.

¹⁰³ Alpha mask is a grayscale image, which defines the transparent areas of a rectangular bitmap thus creating the shape of an object. Z-buffer masks are also grayscale images. The various brightness levels in the Z-buffer image determine the object’s depth. If the Z-buffers are faulty, parts of the object can “bleed” through walls and other objects.



A simple symmetrical object like the chair pictured here is made of only two different bitmaps with the corresponding alpha (middle) and Z-buffer (right) masks. Non-symmetrical animated objects can be composed of dozens of bitmaps. Because *The Sims* can be viewed from three distances, three different sized versions of each bitmap are needed. These can be created automatically.

The simplest custom objects are so-called “recolors”, which only change the coloring of the cloned base object. Some recolored objects have just been re-toned (e.g. from green sofa to blue), while others have been carefully textured with new materials/patterns (e.g. from solid color sofa to striped).

Creating custom-shaped objects is usually considerably more complex than recoloring, as the alpha channels and Z-buffer masks have to be edited also. Fortunately there are plenty of tutorials available. Changes in the shape of objects can be subtle adjustments (e.g. turning a rectangular mirror into an oval mirror) or more major makeovers (e.g. replacing a decorative knight’s armor with a statue of a sitting Buddha). The most difficult objects to customize are those that include complex animations. Instead of four customized bitmaps required by static objects, animated objects might require dozens. Beds, in particular, are notorious for their complexity.

Custom objects on download sites are usually archived according to categories of the in-game Buy mode - i.e. either by the function of the object (Seating, Appliance, Decorative etc.) or by the room type (Kitchen, Bedroom, Outdoors, etc.). Furniture objects especially are often created and shared as matching sets. These sets usually follow a distinct theme (Christmas, Medieval, Oriental), style (minimalist, wooden, etc.) or a color scheme. It is quite common that some central pieces of the sets - such as beds or bathtubs - are only available for subscribers/donators.

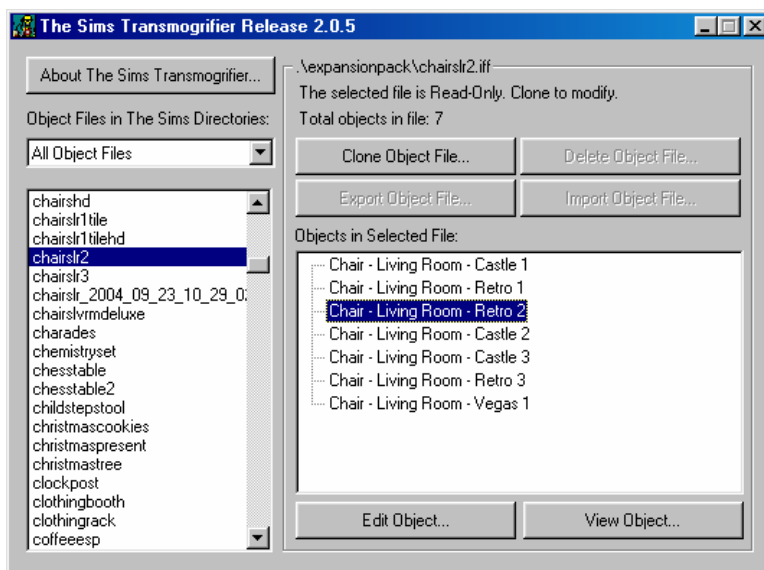


Custom objects are often created in thematic sets (though the objects can also be downloaded separately). Clockwise from top left: a set of original objects “recolored” with a flower pattern; an Oriental bedroom set; furniture from an author who specializes in modern style; an Irish pub set complete with custom skins for the staff; Abandoned farm and Fish farm set from an author who specializes in countryside objects; objects for the “seedy” side of Simsville.

Although Maxis has been very forthcoming about its devotion to *The Sims* modding community, when it comes to custom objects, the official support agenda is seriously lacking. The only object creation tool Maxis has released is ArtStudio, which converts bitmaps into framed paintings. There are no official tools (or support documentation) for creating furniture, appliances, building elements, plants, etc. This is extremely surprising, considering how central house building and furnishing is to the game¹⁰⁴. The lack of official tools, however, has not stopped the fans from reverse engineering their own object editing utilities.

¹⁰⁴ At some point Maxis hinted that one of the reasons they were not releasing an object-editing tool was that the tool would be too complicated for the average Sims fan.

The most important tool for *The Sims* object makers is Transmogrifier (Tmog in the “simmer” lingo)¹⁰⁵. One of the essential features in Tmog is the ability to make clones of existing objects. Object creation for *The Sims* always starts with cloning: an original object with desired attributes (size and behaviors) is sought out and duplicated. The bitmap images that make up the cloned object are then replaced with new graphics and the object’s properties, such as the name, description, and cost customized as needed. If the custom object is based on an original object that came with an expansion pack, others can use it only if they have the expansion pack in question is installed also.

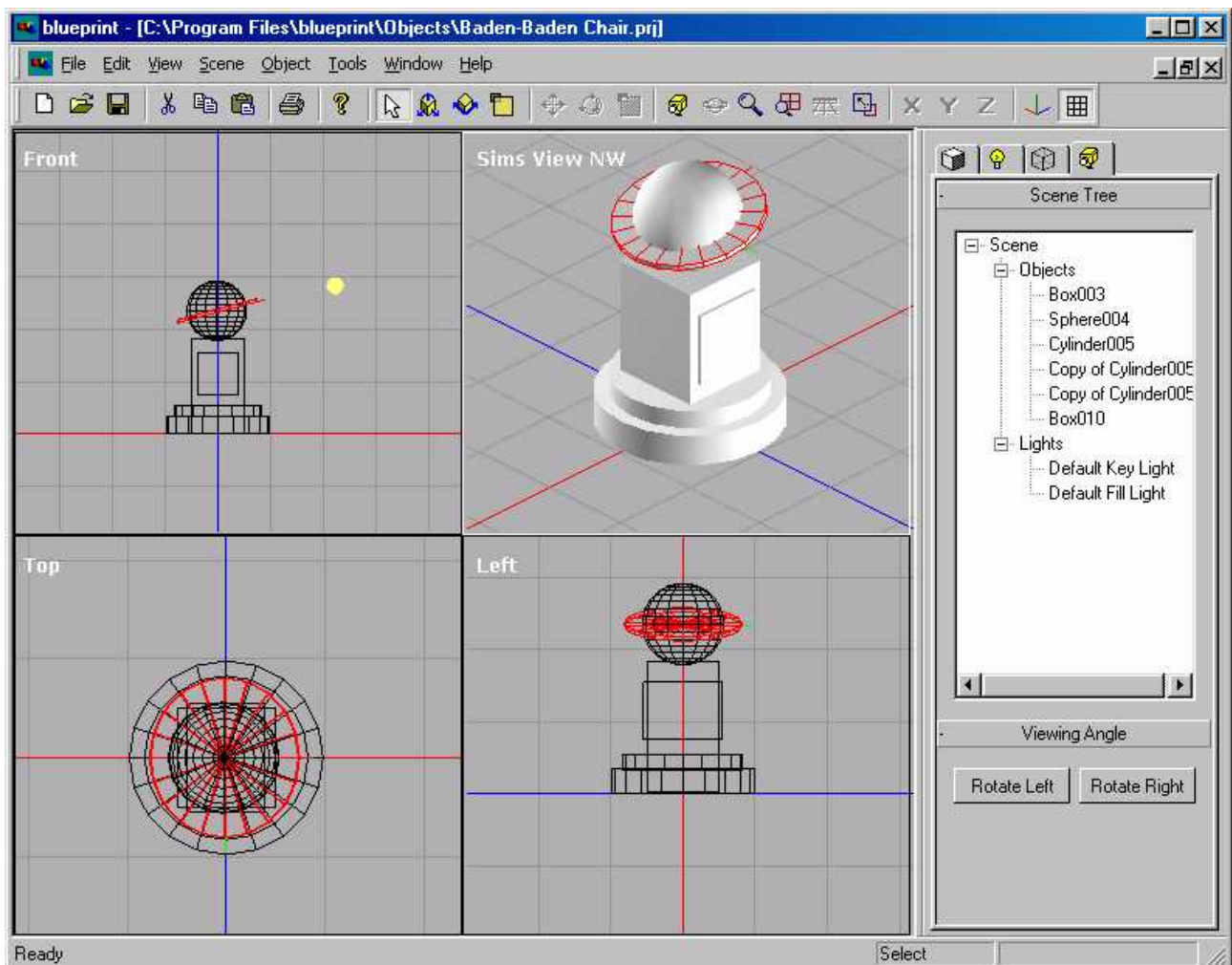


With Transmogrifier Sims modders can clone existing objects and change their graphics to create new custom objects. Each “Tmog” user can register a custom ID number, which is then used when generating the object file names. This prevents conflicts with downloaded objects.

User-created “second-generation” objects can of course also be cloned with Tmog. Web sites that offer custom object downloads have different policies regarding cloning. Some encourage it, some forbid it entirely, and some only allow certain objects to be cloned. Even if cloning is allowed, distribution of the third generation objects may be prohibited. If distribution is also allowed, proper credit for the original author should be given. However, all these policies rely completely on the integrity of fellow fans - there is no way to prevent cloning technically or to “watermark” objects. Unfortunately content theft is a recognized problem in *The Sims* community (and not limited to objects).

¹⁰⁵ Interestingly, Tmog was developed by a former Maxis employee, with apparent cooperation (or at least consent) from Maxis. However, although Tmog is one of the most essential tools in *The Sims* content creation community, there is not a single reference to it on the official Sims site.

As previously noted, objects in *The Sims* are not three-dimensional, but composed of bitmaps taken from four fixed angles. These bitmaps – sometimes called sprites – can be created in several ways. A popular method is to take a photograph of a real life object and scale/skew it appropriately in a graphics application. While this method works surprisingly well, best results are usually achieved when the object is actually modeled in a 3d application, and then rendered from four angles to create the object bitmaps. This approach gives object makers more creative freedom and produces objects that blend naturally into the game world.



Although any 3d application is suitable for object modeling, a simple and free Sims-specific 3d utility called Blueprint is also available¹⁰⁶. The modelling features of blueprint are very basic, but it can edit various object properties and export "ready-to-play" Sims objects.

Installing custom objects is very simple. All the user has to do is to download and unzip the files into the right folder and they will be available next time the game is loaded. However,

¹⁰⁶ The author of blueprint is a professional programmer who has provided *The Sims* community with several popular tools. Like the author of Transmogrifier, he has close ties with Maxis and has received assistance for his efforts.

locating the downloaded objects from the in game catalogues can be difficult, since passionate Sims fans can have hundreds of custom objects installed simultaneously. Fortunately there are fan-made object management utilities such as SimCategorizer.

4.3.4. Hacked objects

As mentioned before, most of the objects in *The Sims* have some interactions associated to them. When the object is clicked, a pie menu opens and lists the available interactions. A couch object, for example, has "Sit" and "Nap" interactions, while a computer object has "Play games", "Find a job" and "Turn on/off" interactions. When selected, each interaction triggers a different animation sequence and has a different effect on the sim's well-being. Objects whose functionality has been edited are known as "hacked objects".



When an object is clicked, a pie menu pops up listing the available interactions. The sofa has Sit and Nap interactions, which both trigger different animations and have different effect on the sim.

While hacked objects are created for various purposes, the most common motivation for using them would seem to be storytelling. Hacked objects give the player more control over *The Sims'* behavior and enable interactions (and animations) that would not normally be possible. There are, for example, hacked objects that can be reserved for a certain sim so that no one else will try to cut in. Lockable doors also help to keep unwanted intruders from ruining the set-up. Some objects are hacked so that *The Sims* will continue to use them until the player releases them. This gives the player more time to set up the scene/screenshot. Teleportation objects, which spawn a chosen character to the location of the object, allow players to quickly populate the scenes with appropriate cast.

Even more empowering for storytelling purposes (but also fun in their own right) are hacked objects with custom animations. Each behavior attached to an object triggers a pre-scripted animation sequence, and with fan-made hacking tools these sequences can be changed. There is for example a mirror object, which, instead of triggering the default “practice speech” animation, triggers a custom-made ballet dancing animation. Floor tiles usually do not have any functionality, but one object hacker has created a popular series of “gym tiles”, which trigger various exercise animations (sit-ups, hand-stands, push-ups, stretching). Although most hacked objects are perfectly decent, it is inevitable that object hackers have created beds that trigger x-rated animation sequences. (There are also hacked sex toy objects.)

Of course not all hacked objects are created or used for storytelling purposes. Perhaps the most common types of hacked objects are objects, which replenish sims’ need and/or skill levels quickly. On a regular recliner chair it might take half a day of game time to completely restore the sim’s energy level, but on a hacked recliner it happens in minutes. There are rejuvenation chambers which will restore all need levels immediately. In fact, many hacked objects could be considered cheats – assuming cheating is even possible in a game like *The Sims*.

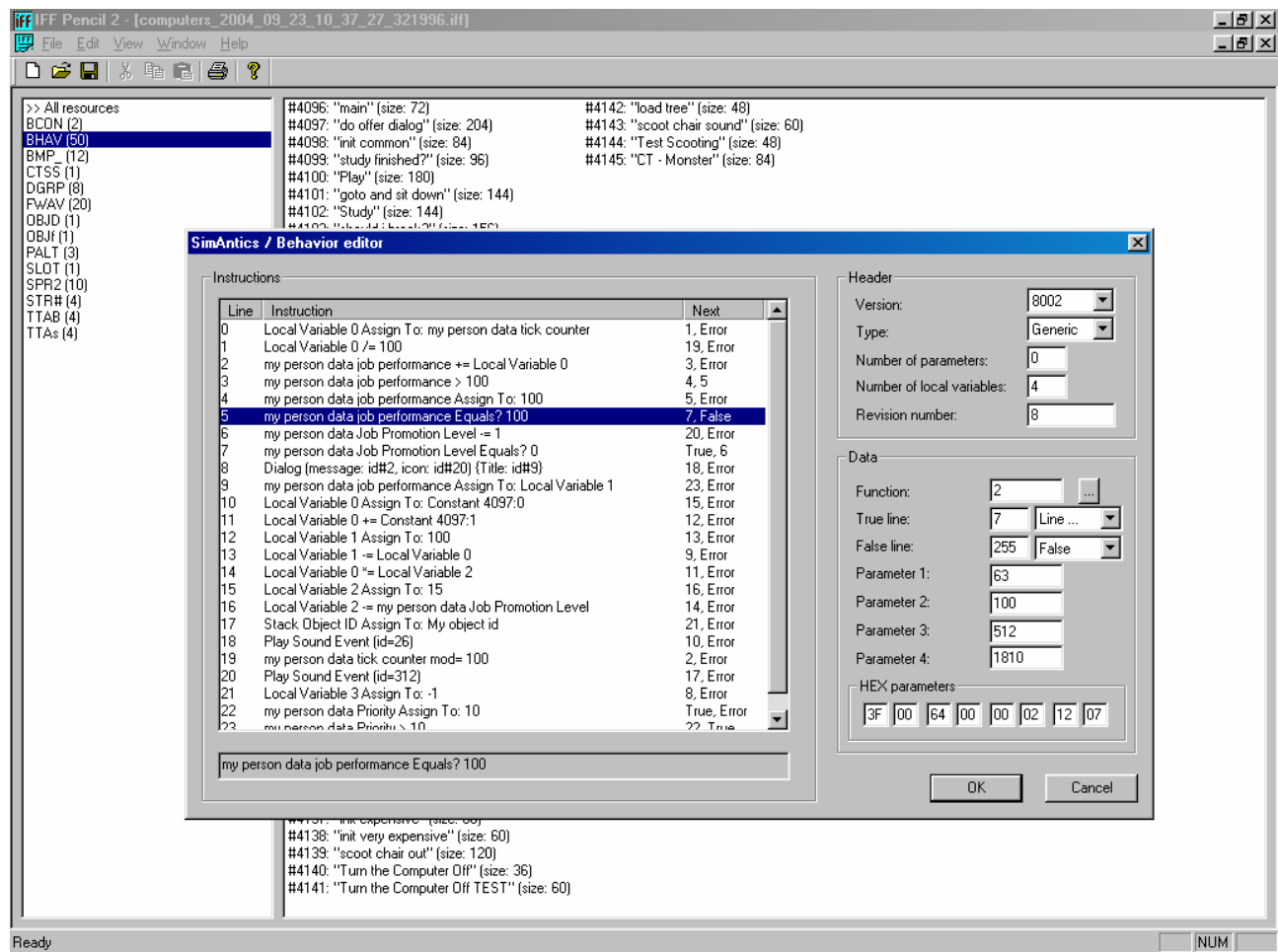
There are also plenty of hacked objects that are meant to be appreciated on their own merits. An impressive example of such an object is the Slice City.¹⁰⁷ Slice City is a power plant object (based on a cloned plant object, appropriately), which, when placed on *The Sims*’ lot, will gradually spawn a whole city, very much like the one in Will Wright’s original SimCity game. There are even tiny citizens (hacked cockroach objects) bustling around the buildings. Sims must take care of the city in order for it to expand. If the city is neglected, a disaster will eventually wipe it out. However, *The Sims* can also crush the citizens and destroy the buildings themselves.

¹⁰⁷ So impressive that even BBC News has covered it: <http://news.bbc.co.uk/1/hi/technology/3495285.stm>.



These few examples give an indication of the diversity of hacked objects. From top left: martial arts mat, ballet mirror, aerobics trampoline and handstand tile; a Santa Clause NPC who gives gifts to good kids and ashes to bad ones; a NPC maid who always stays in the house and does not cost anything; curtains that automatically open in the morning and close at night; a hen that lays eggs (if nurtured), which hatch into chickens, which grow into hens, which lay more eggs; a climate control machine (it can even rain cats and dogs); and Slice City, a power plant object that spawns a SimCity like metropolis on *The Sims'* lot.

The tools for object hacking are naturally all fan-made. Pioneering object hackers used hex editors and a tedious trial-and-error approach. Eventually, with persistent reverse engineering, Sims fans were able to produce dedicated hacking utilities with more user-friendly graphical interfaces. First such tool was Script Station developed by a Dutch modder. Maxis warned the author of a possible copyright infringement suit at first, but the case was never put forward and Script Station has been openly available at various community sites. However, nowadays most object hackers prefer the more recent, robust and versatile IFF Pencil 2 utility from the same author (some use both).

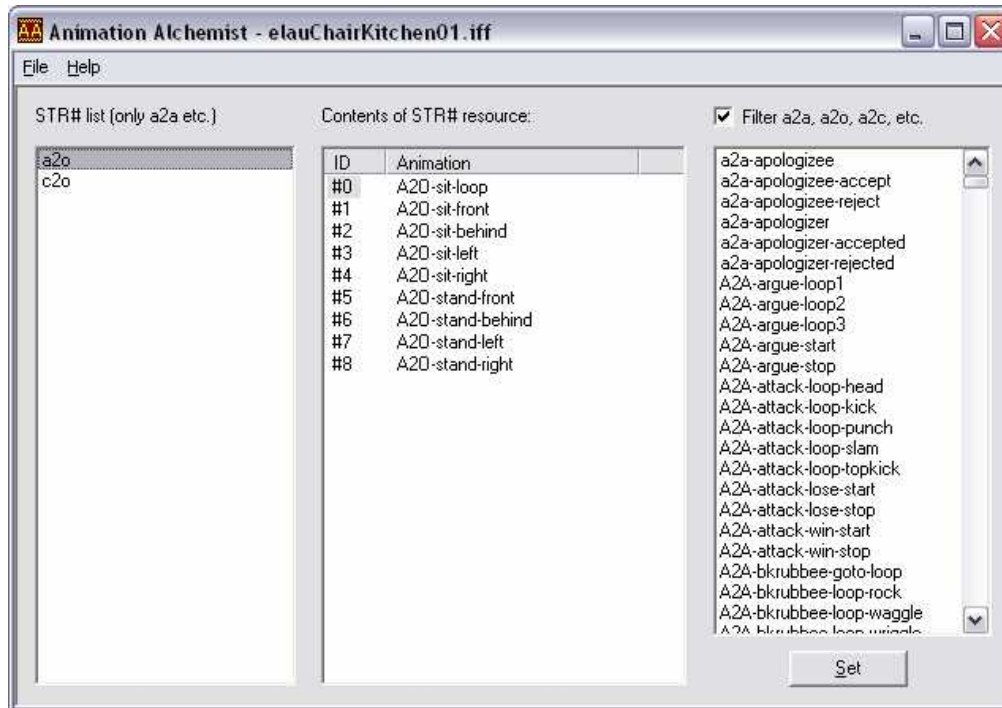


Practically all parameters related to objects can be tweaked with IFF Pencil. However, it is well nigh impossible to create anything sensible with the tool just by “playing around” with the values.

Although IFF Pencil 2 is certainly more accessible than a basic hex editor, the learning curve is still mighty steep. An intricate understanding of the proprietary scripting language (“simantics”) and object file format is necessary. The tool itself comes with practically zero documentation, and hacking tutorials overall are few and far between.¹⁰⁸ Animations assigned to object behaviors can be edited with hacking tools, but there is also a dedicated tool called Animation Alchemist for the purpose. Animation Alchemist cannot create completely new animation sequences, it can only reassemble existing sequences in new ways. Luckily there are hundreds of sequences available in the original *The Sims* - a hugging animation, for example, consists of six separate sequences - and even more with expansion packs. Completely new animation sequences can also be created with a tool called SimPose. As the name suggests, the main function of SimPose is

¹⁰⁸ Things are not totally hopeless for novice hackers. There is a dedicated hacking forum at <http://forum.hackersresource.info/>. At the time of writing an entry level “hacking class” is starting at the Simantics Online School discussion forum (<http://forum.simantics.com/>).

to allow skin/mesh makers to create desired poses (imitating those seen in fashion catalogues) for their skin/mesh screenshots. The poses can then be connected into keyframe animations in SimPose and attached to an object with Animation Alchemist.



With the fan-made Animation Alchemist the existing animation sequences can be reassembled and attached to custom objects.

4.3.5. Walls, floors and roofs

Walls, floors and roofs are patterned bitmaps that can be applied to corresponding surfaces in the game's Build mode. Wall textures are used for both interior and exterior walls. In addition to continuous patterns, it is also possible to create murals, which consist of few adjacent wall elements. Floor textures are used for interior floors, but also for driveways, patios and such.

Of all custom content types, walls, floors and roofs are the easiest to create. The bitmap graphics can be created with any imaging software. Roof bitmaps are ready to be used as such, but walls and floors need to be converted to *The Sims*'s file format with the official Home Crafter tool. Home Crafter was one of the very first content creation tools Maxis released (it was out months before the game) and seems to be the rare official tool that fans are more or less content with - at least no one has bothered to create an unofficial alternative. Home Crafter is indeed very user-friendly, and offers a nice interactive preview feature where the custom walls/floors can be tested.



An assortment of walls and floors (from top left): seventies-style wallpapers, Egyptian walls, assorted tapestries, floor tiles.

Although not quite as plentiful as skins and objects, there are nonetheless thousands of custom walls and floors available on download sites, ranging from solid color tiles to fancy patterns edited from photographs. Quite often several wallpapers based on the same design motif are created, each with a slightly different color scheme. It is also common to have matching custom floors offered with the custom wallpapers. Besides being available as separate downloads, wallpapers and floortiles/carpet are also often included in furniture object sets to complement the chosen room style.

4.3.6. Lots

Lots are the gameplay environments of *The Sims*. The whole gameworld is divided into neighborhoods, which in turn are divided into fixed sized sections called lots. As downloadable

custom content items lots include both the land area and the buildings constructed on it. There are two basic types of lots: houses and public lots. Houses are the living quarters of *The Sims*. They are built on the neighborhood areas where one sim family always occupies one lot. Public lots, on the other hand, are built on various community areas. They are places where sims can only visit: malls, parks, film studios, holiday resorts, etc. (All the community areas have been added to the game with different expansion packs.)

Lots are constructed in the game's Build mode. (There are no unofficial external tools for the task.) The player can either choose to remodel an existing lot - both the landscape and the buildings - or start from scratch on an empty lot. The lots can be constructed entirely of the original game items, but all the installed custom objects, walls, and floors can of course also be used. However, using custom content makes it a little more complicated to share the lot with others. If only original content is used, the whole lot can be shared as a single small file that only includes references to the components used. If custom items are used, they have to be downloaded and installed separately. Sometimes small custom items such as walls and floors are included in a zip file, but sometimes only a "shopping list" is provided with notes about the requisite separate downloads (these can be from other web sites). Most lot makers seem to settle for the official content. Of course, objects from the expansion packs are used extensively. These lots require that the users have the same expansion packs installed as the author.

The game mechanics of *The Sims* set some limitations to architectural designs (e.g. buildings can only have two floors), but within these limitations fans have been able to create very impressive constructions. Fan-made houses range from modest "first homes" to extravagant mansions with pools, gardens and various annexes. Some houses pursue a distinct look, such as a medieval castle, Mediterranean condo, or countryside home, while some attempt to recreate houses seen in television shows. Various famous buildings such as the White House can also be downloaded to *The Sims* neighborhood. Of course some houses are just "dream houses" of their authors. The most extravagant houses may be difficult to manage in actual gameplay, but they provide a lot of things for *The Sims* to do and are therefore interesting backdrops for storytelling. Naturally they also give the authors a better chance to flex their creative muscles.

The management is not an issue with public lots, since *The Sims* can only visit them. Public lots can be constructed from the same items as houses, but there are also items designed for specific public lots, such as restaurants, shops, night clubs, hotels, beach parks, etc.

Besides the pre-built lots, fans also create and share "landscaped lots", though these are rather rare. Landscaped

lots have no buildings, but the yard has been extensively reworked with added hills, plants, ponds, streams, and so on.



Different types of lots (from top left): a residential mansion; a recreational downtown lot with shops and restaurants; a modern mansion built and furnished almost entirely with custom objects on an extensively landscaped lot; empty landscaped lots from custom nature objects.

As already said, the installation of lots is usually very simple. Assuming the lot uses no custom items, the lot file only needs to be copied to the proper folder, after which it can be imported from within the game. Installation from the official Sims Exchange site is even simpler: when downloaded with Internet Explorer browser, the lots are installed automatically while the game is running. The lot sections on neighborhoods and community areas are numbered, and each custom lot available for download is assigned to a certain “address”. When installed, any previous buildings on the particular section will be erased. With fan-made EliSims utility it is possible to move houses from one neighborhood to another and free up sections for new houses without having to give up any of the existing ones. EliSims also allows backing up whole neighborhoods.

4.3.7. Tools

Because content creations is such a central aspect of *The Sims* experience, it is difficult to draw a distinction between modding tools and any other utilities fans may use to tinker with the game. Some of these tools, both official and fan-made, are discussed in the following.

All in all Maxis has released ten official tools for *The Sims*. All of these except for The Sims Creator skin editor are available for download at the official site. However, for most tasks fans generally prefer unofficial utilities to the official ones. Though generally very user-friendly, Maxis's tools lack the versatility of unofficial tools. They also have not been updated to support the later expansion packs. One of the few official utilities still in active use is The Sims File Cop. It's a simple validating tool that scans through *The Sims* data folders and automatically removes all corrupted custom content files (skins, objects, walls, floors).

The most comprehensive listing of unofficial Sims tools - over hundred in total - can be found at The Sims Workshop web site. In addition to the content creation tools - many of which have already been discussed - there are numerous content management tools for importing/exporting, previewing, selecting, and organizing custom content. There are also tools for automating the installation of downloaded files, for validating downloaded files, for removing duplicates, and for packing individual downloads into more efficient archive files.¹⁰⁹

The abundance of content management utilities is understandable, since an enthusiastic *The Sims* fan can easily have hundreds or even thousands of custom content files installed on his computer - not to mention all the official goods from the numerous expansion packs. *The Sims* fan community uses the term "download junkie" to playfully refer to fans who like to collect every piece of custom content they can find. In addition to the external file management, many unofficial utilities also allow users to manage custom content in game, i.e. categorize buyable objects and skins. It should perhaps be noted that there is also an official content management utility called Sims Hot Date Object Manager, but most fans seem to regard this tool as worthless.

Because of the nature of *The Sims*, some unofficial tools can be simultaneously considered as cheating and content creation tools. SimEnhancer 3d, for example, allows users to edit practically every attribute of *The Sims*, from their appearance (i.e. skin) to their wealth, skills, relationships, personality traits, etc. Clearly these features can be used for cheating; it is

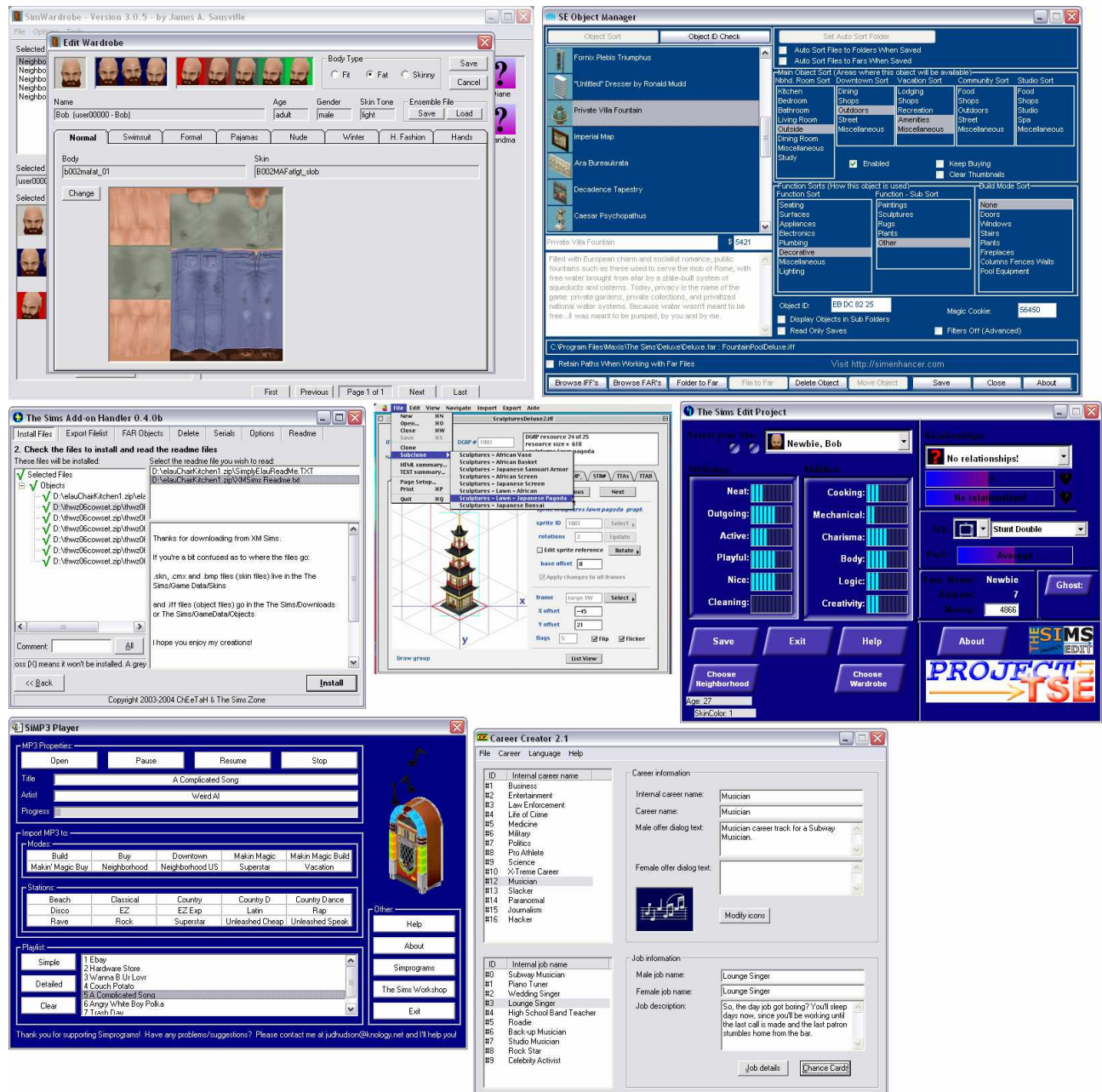
¹⁰⁹ Load times for *The Sims* increase rapidly as new expansion packs and custom content is installed. It can take more than half an hour to load the game. Packing individual objects and skins into archives usually speeds things up considerably.

possible to give a sim maximum skills and unlimited wealth with few clicks instead of spending months playing the game and building the character properly. On the other hand, since *The Sims* does not have any real goals, the concept of cheating is somewhat vague. SimEnhancer 3d (and other similar tools, like *The Sims Edit*) allows users to create exactly the kind of characters they need for playing out certain kind of scenarios. This is especially useful for storytelling purposes. It is worth to note that, although developed by an individual fan, SimEnhancer 3d is a commercially sold application. Although it is one of the few that are not free, it does further underline the commercial streak in *The Sims* modding scene.

Finally some miscellaneous tools are worth mentioning just to give a better picture of the diversity of unofficial tools. Career Creator allows users to edit *The Sims'* career paths (names, salaries, skill requirements, etc.) or replace them with completely new ones. Simsky is a tool for editing the Family album screenshots and incorporating poses created in SimPose into photo backgrounds. SiMP3 Player is one of the utilities that allow users to play their own mp3 music selection in the game. ChangeIT! customizes the TV screens seen in the game. With SimTeen players can turn kids into teenaged sims (basically thin adult sims with children's heads) and put them through a high school career path.

The tools mentioned above have all been PC applications. *The Sims* has also been very successful among Macintosh users, and there are plenty of unofficial Mac tools for creating and managing custom content. Maxis, however, has been surprisingly lazy in servicing Mac users. SimTech website¹¹⁰, maintained by a couple of seasoned programmers, has been an important information resource for all tool developers, but for Mac developers in particular. It offers impressively detailed documentation of the file formats in *The Sims*.

¹¹⁰ <http://simtech.sourceforge.net/home/welcome.html>



Some examples of the extensive tool line-up produced by *The Sims* fan community (from top left): Sims Wardrobe (for skin management); Object Manager (a commercial object management utility); Add-on Handler (automates custom content installation and checks for errors); IFF Snooper (a versatile object editor for Macintosh computers); Sims Edit (for tweaking various character properties); SIMP3 Player (for playing personal music files in game); Career Creator (for editing and creating career paths).

4.4. Summary and notes

The Sims, released in 2000, was such a unique combination of game mechanics that it created a genre all its own. Partly a virtual dollhouse, partly an architectural toolkit, *The Sims* also became a huge crossover success: it has been estimated that more than half of the millions who bought the game were female.

Though numerous expansion packs have kept the franchise on store shelves for years, the continued success of *The Sims* also owes immensely to the active modding scene it has spawned.

The Sims is a prime example of lead designer Will Wright's desire to encourage player creativity. User-created content is at the very heart of *The Sims* experience: not only are character creation and house building central game mechanics, but the game design also makes it easy to import various types of custom content. To further encourage fan production, Maxis has released numerous official content creation tools. These are offered freely from the official community site, where fans can also share their creations.

Though the official community site and its discussion forums are very popular, it offers surprisingly little support for the fan producers. Fortunately unofficial communities fill this void. The biggest unofficial community is The Sims Resource. Like is the case with practically all Sims fan sites, the main feature of The Sims Resource is the massive custom content archive. And like so many other Sims file archives, the archive at Sims Resource requires a paid subscription for full access. This kind of commercialism is a distinguishing feature in *The Sims* community. The biggest custom content archives are generally subscription based, while smaller sites often offer their best content only for those who have donated some money for the site's maintenance. The commercial streak extends to "team sites" where small groups of modders publish their creations, as well as to the personal sites of the more renowned modders. Smaller fan sites often concentrate on a certain custom content type (such as objects) or a theme (such as the Orient).

The custom content that *The Sims* fans create consists only of small, individual add-ons. Maxis has not released any source code for the game that would allow more extensive modifying of the game mechanics. The content types are skins, head and body meshes, objects and hacked objects, walls and floors, and lots. Skins are the most popular type, and also the content type where the official support is most extensive.

Fans have also produced a wide range of unofficial utilities to fill up the rather peculiar omissions in the official tool line-up. The biggest disappointment for fans has been the lack of object editing tools. This is a significant shortcoming: not only are objects used both for building and furnishing the sims' homes, they also trigger most of the game's interactions. Another area where official tools are sorely missed is content management. There are no official utilities, though self-proclaimed "download junkies" can have thousands of custom content items installed. Fortunately the fan community has been able to produce versatile tools for both object editing and content management.

The tools Maxis has released have generally been user-friendly, but their limited features and lack of updates have been widely panned by the fans. Consequently, fan-made tools

are usually preferred even for the tasks where official tools are available.

4.4.1. Notes

One of the most distinguishing features of *The Sims* modding scene is the explicit commercialism. This is very unique in the modding culture, which generally holds the freeware philosophy in high esteem. (Though there are few commercially released mods for *Half-Life* and other first-person shooters, even these mods are usually also available as free downloads.)

The fact that commercialism exists in *The Sims* scene suggests two things. First, that *The Sims* scene is somewhat isolated from the traditional modding culture. This is quite understandable, considering the very different demographic of *The Sims* and other heavily modded titles. The second thing the commercialism of *The Sims* scene suggests is that using and sharing custom content is a more mainstream aspect of *The Sims* experience than it is for other games.

For the other games studied here, *Half-Life* and (especially) GTAs, the usage of mods and other custom content is still mainly limited to more hard-core gamers (retail mods like *Counter-Strike* notwithstanding). In the case of *The Sims* it is likely - though this has not been investigated - that a vast majority of players download custom content. Even with the small files sizes of Sims custom content, this means that the traffic on popular fan sites is enormous: the FAQ on a popular SimFreaks site claims the monthly hosting fees were approaching \$4,000 when the administrators decided to switch to a subscription-based service. In cases like this charging for the content is the only viable solution for survival. And since the survival of these sites is, naturally, in the best interests of EA/Maxis, the commercialism has not been interfered with.

Another thing that distinguishes *The Sims* modding scene from the previously discussed is the lack of collaborative projects. There is collaboration on a smaller scale - help is readily offered at the forums, modelers share their meshes with skin artists, etc. - but more extensive team/community projects are lacking. Obviously this state of affairs is related to the fact that the output of *The Sims* fans consists only of small-scale modifications. There is no need for modding teams when creating something like objects and skins.

However, one could look at this from another angle also: is *The Sims* community output small-scale because of the lack of collaboration? If a group of talented skimmers, modelers, object makers, and object hackers pooled their talents on a common project, would it be possible to produce, for example, a whole unofficial expansion pack?

Alike *Half-Life* modders, *The Sims* modders have received extensive support from the developer. As has been mentioned, the guiding approach to game design for the lead designer Will

Wright is to support user creativity. Wright has actually admitted to having been particularly inspired by the FPS modding communities.¹¹¹ Of course, the support agenda of Maxis has been quite different from that of Valve. Whereas Valve obviously caters to the hard-core modders - making any sensible use of the Hammer editor or the SDK requires a lot of dedication - Maxis has attempted to make modding accessible even to the less computer-savvy fans. And, with its simple one-task tools and easy integration of custom content, Maxis has succeeded remarkably well.

Unfortunately, however, simplicity seems to have come at the expense of versatility. Although Maxis has released ten official content creation tools, there are no tools for creating custom character models, animations, interactions, sound effects or even objects. Sorely missing are also official content management and/or preview tools. As for the game itself, it is frustratingly prone to crash if there are even minor errors in the custom content files. Expansion packs have brought along all sorts of confusions and compatibility issues. All in all it should be apparent that while *The Sims* certainly holds up to Will Wright's philosophy and does encourage and support user creativity exceptionally well, there is still plenty of room for improvement. Some fans have suggested that perhaps Maxis - or rather EA - has intentionally limited the level of modding support, so that fan-made free content would not jeopardize the sales of the official expansion packs.

It is interesting to see if and how these shortcomings will be addressed in the upcoming *The Sims 2*. At the moment things look promising enough: the character editing suite Body Shop, released months before the game, is much more versatile than any official tool for the predecessor. The official Exchange web site has been expanded, and now also covers custom objects and "machinima" videos made with *The Sims 2*. The official file exchange service will be especially important, since the file sizes for *The Sims 2* custom content are much bigger than for its predecessor. The sequel also includes much improved content management options built in the game.

In any case the transition from *The Sims* to *The Sims 2* will be challenging for both Maxis and *The Sims* modders. Because of complete technological overhaul, none of the tools or work methods the community developed for the original game work with the sequel. Not only will modders have to start the learning process again from scratch, but the whole creative process has also become considerably more demanding. *The Sims 2* is completely three-dimensional and much more detailed than its predecessor. It is no longer possible to create objects from images found on Google, and celebrity skins copy-pasted from

¹¹¹ (Au 2002).

photographs will look horrible close-up. In a very real sense modding *The Sims 2* resembles modding any other 3d game. If Maxis will again succeed in making custom content creation accessible to the average fan, the whole modding culture could be taking another major leap towards mainstream.

5. Case Study: Grand Theft Auto III & Vice City

5.1. Description of the game(s)

The main character of the 3rd person driving/action game *Grand Theft Auto III* (Rockstar Games, 2002) is an anonymous, unspoken thug. Fans have christened him “Fido”, after a nickname used in a cut scene. Recently escaped from prison, Fido’s objective is to take revenge on the people who set him up, and at the same time climb up the underworld hierarchy of Liberty City. This is accomplished by completing various odd jobs (missions) for local crime lords and gangs: chauffeuring mistresses, assassinating enemies, rescuing prisoners, destroying vehicles, collecting drug deliveries, etc.

Fido gets around the huge Liberty City either by foot or by hijacking cars off the streets. There are fifty different vehicles in total, ranging from family wagons and sports cars to fire trucks and tanks. The cars have different strengths (speed, durability, etc.), which must be taken into account when attempting different types of missions. As the game progresses the player can purchase increasingly powerful weapons, but the missions will also get progressively more difficult. Most missions can be completed with various different methods, accentuating the feeling of freedom in the game world. In addition to the main storyline missions player can take on numerous side quests: race cars, earn money as a cab driver, search for hidden packages, attempt car stunts (jumps from ramps, hills, roof tops etc.), and even fight crime as a police. Nine radio stations with diverse musical offerings (opera, reggae, talk-radio, etc.) provide the game’s soundtrack.



Though the basic design and technology of *Grand Theft Auto III* (left) and *GTA: Vice City* (right) is almost identical, the radically different setting and a host of new features, such as motorcycles, made *Vice City* a worthwhile sequel.

The release of *GTA3* (initially for PlayStation 2 console in 2001) was met with quite a bit of controversy. While the gaming community unanimously praised the scope and liveliness of *GTA3*'s 3D universe and the open-ended variety of gameplay¹¹², public debate quickly focused on the decadent world-view the game supposedly celebrated. In some countries the game was banned altogether, in others it fuelled the ongoing discussion about videogame violence in general.¹¹³ Amidst the controversy, *GTA3* went on to win numerous game-of-the-year awards and became one of the best selling videogames of all time. Six months after its debut on PS2, *GTA3*'s success continued with the release of a PC version in the spring of 2002. Apart from a slight graphical update very little was changed. Fans who had hoped for online/multiplayer capabilities were disappointed.

The sequel *Grand Theft Auto: Vice City* (Rockstar Games, 2003) was essentially the same game in a new environment. Again the player starts out as a small time thug - though this time he has a name (Tommy Vercetti) and a voice (actor Ray Liotta) - whose main objective is to rise to the top of the underworld ladder by doing dirty work for various crime bosses. Though VC expanded the open-ended playfield even further with new vehicles (motorcycles, helicopters), interior locations, and property management, the most distinct feature of the sequel was the new setting. Whereas the urban Liberty City of *GTA3* mirrors present day New York, the sunny and neon-colored *Vice City* of the sequel was a recreation of 80s Miami. Much inspiration was drawn from the classic television series *Miami Vice* (USA 1984-1989) and the gangster movie *Scarface* (1983). The period atmosphere was completed perfectly with an exhaustive soundtrack of 80's music, ranging from Michael Jackson to Iron Maiden and Run DMC. *Vice City* more than lived up to the hype surrounding its launch and went on to become even more successful than its predecessor. (The controversy around the series continued. This time the hot topic was the game's portrayal of ethnic minorities, more specifically Cubans and Haitians¹¹⁴.) The PC version was released in the spring of 2003, again with little changes and no online connectivity.

Grand Theft Auto: San Andreas, a conclusion to the "trilogy" started by *Grand Theft Auto III* and *Vice City*, will be released

¹¹² The open-ended gameplay of *Grand Theft Auto III* has become heavily imitated in the wake of the game's success. See for example *Simpsons Hit & Run* (Radical Entertainment 2003), *True Crime: Streets of LA* (Luxoflux Corporation 2003), *Jak II* (Naughty Dog 2003). See <http://ps2.ign.com/articles/528/528543p1.html?fromint=1>.

¹¹³ <http://www.usatoday.com/tech/techreviews/games/2001/12/20/game-glorifies-crime.htm>

¹¹⁴ http://www.hispanicprwire.com/news_in.php?id=1874&cha=10

for the PlayStation 2 console in the October of 2004. The PC version will most likely follow in the spring of 2005.

5.2. Overview of the modding scene

Note: GTA3 and Vice City share a common modding scene, which in the following discussion will be referred to as the GTA modding scene. The games will be referred to as the GTAs. It should be noted, though, that the previous installments of the GTA series - Grand Theft Auto (1996) and GTA2 (1999) - have also been quite actively modified, but are not a part of this study.

The most distinguishing feature of the *GTA* modding scene is obviously the fact that it operates completely without developer support. There are no official modding tools or communities, and no technical information regarding the game architecture has been released. The only concession to customization was to allow player to change the look of the player character with skins and listen to personal MP3 collection as a soundtrack. Rockstar does not condemn modding in general; when the first games in the *GTA* series were released, Rockstar was extremely supportive, releasing documentation and tools to the community. However, the new generation of *GTAs* utilizes a game engine based on third party RenderWare platform, and Rockstar obviously does not have the liberty to publish any technical details of the engine, let alone tools to "hack" it. The promotional sites for the PC versions of *GTA3* and *VC* do have links to fan sites that offer custom content downloads.

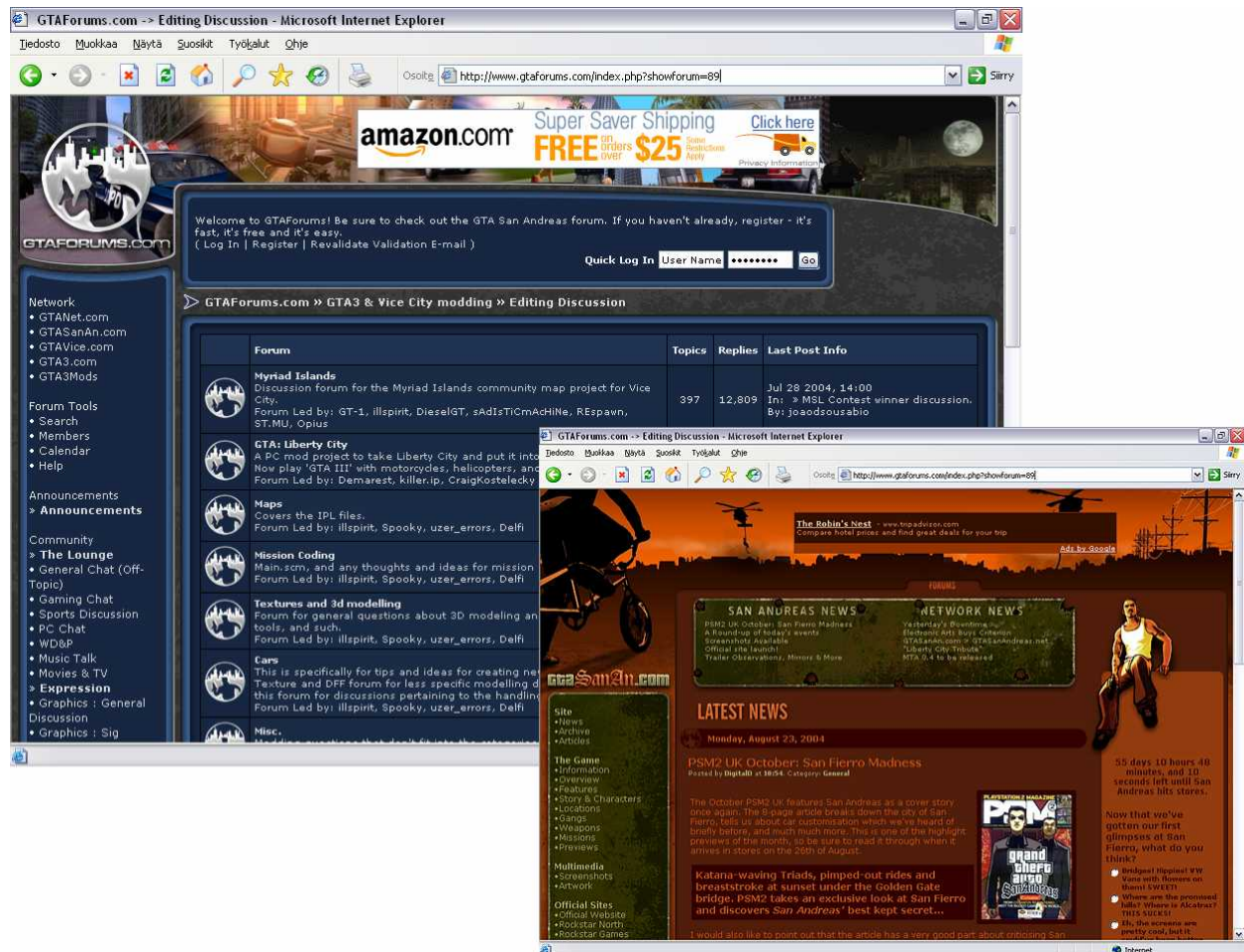
The lack of official support may have hindered the development of the *GTA* modding scene, but in the end has not prevented it from blossoming. Pooling the efforts and knowledge of the huge fan base via the Internet, players soon figured out how *GTA3* works and how it could be modified. Some lucky (or unlucky, depending on the perspective) information leaks regarding the game's RenderWare engine helped quite a bit. Soon enough, simple modifications for *GTA3* started to circulate on the Internet alongside crude unofficial editing tools. By the time *VC* was released almost every aspect of the game could already be modified. The accumulated *GTA3* modding knowledge could mostly be applied to *VC* with minor adjustments, though there have also been some frustrating issues with the updated RenderWare platform. Currently most *GTA* modders seem to concentrate on *VC*, probably because of its more diverse mechanics, although *GTA3* also still has its share of devoted fans. There is actually a mod that ports the whole city of *GTA3* to *Vice City* engine (more of this below).

While it is impressive how far *GTA* fans have taken the modding scene all by themselves, they have not been able to completely overcome the problems associated to lack of official support. As *GTA* mod projects have gotten more ambitious, modders have run into increasingly frustrating technical hurdles. The worst part is that some of the features modders would like to tweak have been "hard-coded" to the game executable, which is explicitly off-limits for modders. This has prevented *GTA* modders from realizing radical total conversions like the ones that dominate the *Half-Life* modding scene. The mod-unfriendliness of the game design also means that using modifications is often cumbersome and off-putting to a casual gamer (although modders have created utilities to simplify the process).

5.2.1. Organization of the modding scene

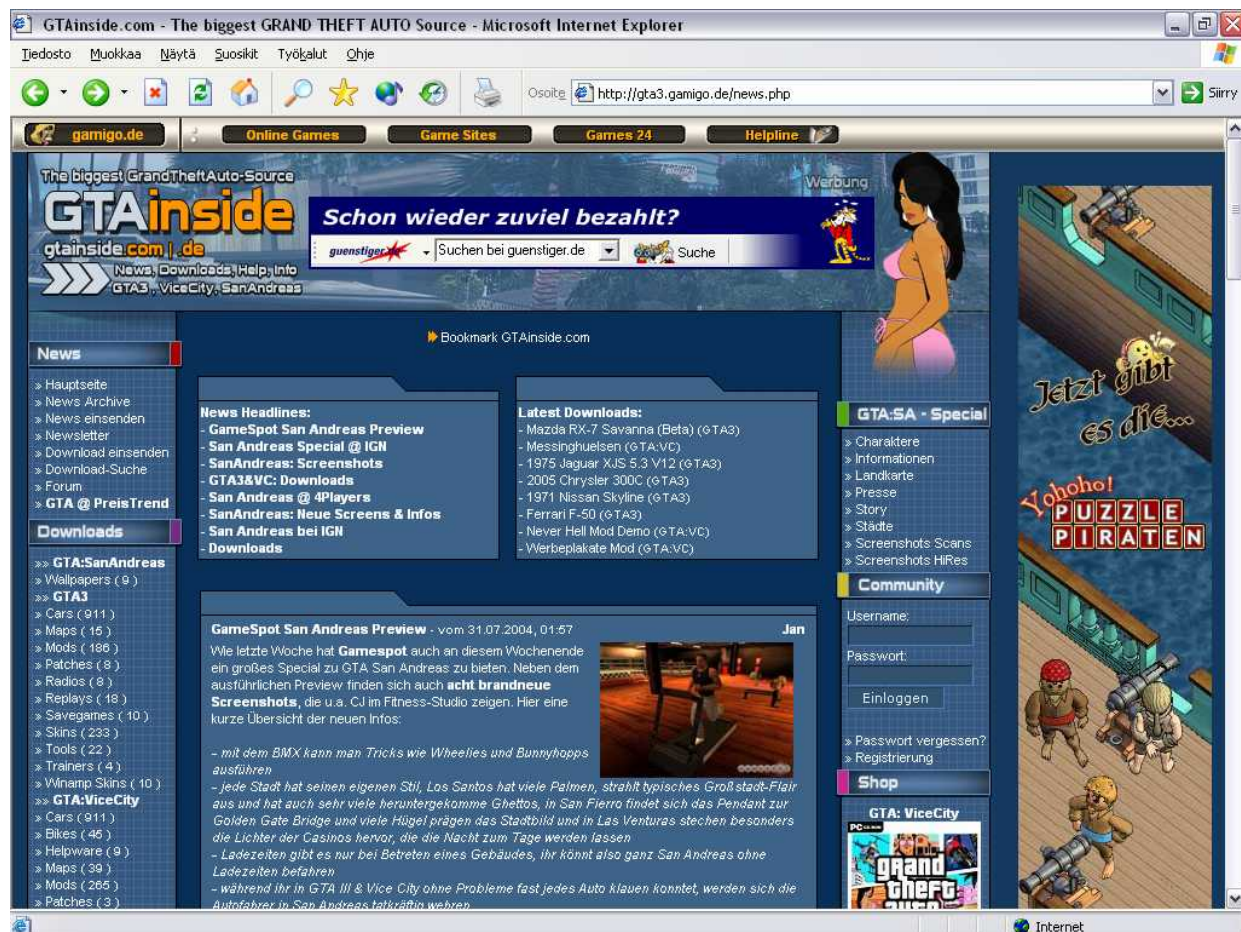
As already mentioned, Rockstar does not host any official fan communities for the *GTAs*. The promotional sites have links to few unofficial fan sites, though the links have not been updated since the games were released. Because of the games' success the fan base for *GTAs* is obviously enormous and able to support several very active unofficial communities. The biggest of these is the GTAForums discussion forum with over 40 000 registered members. Although GTAForums covers all aspects of *GTA* fandom, its meaning to the modding scene is especially significant. Practically all of the experienced "masters" of the scene are regulars, and many of the technical breakthroughs in *GTA* modding have come about through their collaboration on GTAForums. The most ambitious *GTA* "total conversion" projects are also developed via GTAforums.

GTAForums is a part of the *GTA* Network, which also hosts separate fan sites for all the *GTA* games, as well as a modding resource called GTA3Mods. Of the separate fan sites only the one dedicated to the upcoming *San Andreas* is actively updated, and the modding site has also been abandoned quite some time ago.



With over 40 000 registered members GTAForums is by far the largest *GTA* fan community. It is also the epicenter of the *GTA* modding scene. Sister site *GTA San Andreas* provides daily news and any imaginable information related to the upcoming sequel.

Despite its central role in the scene, GTAForums is of course not the only site where *GTA* fans congregate and modders collaborate. Some of the other popular fan communities are GTAGaming, Planet Grand Theft Auto and Codename: *GTA*. These are all general fan sites and serve the whole *GTA* fan community, not just modders. In addition to discussion forums, they provide gameplay info, screenshots, review links, *GTA*-related news, etc. Each of the sites also has an extensive custom content file archive. Planet Grand Theft Auto, which is a part of the Gamespy Network (like Planet *Half-Life*) naturally links to Fileplanet's archives. Overall, however, general file hosting services such as Fileplanet are not as important to the *GTA* modding scene as they are for the *Half-Life* modding scene. As will be discussed later, *GTA* modders very rarely produce large total conversion mods that would require excessive amounts of bandwidth.



Though this study concentrates on English language sites, the biggest custom content file archives can be found on German community sites such as GTAinside.

Although there are plenty of general *GTA* fan sites, there are not any sites dedicated entirely to modding. And while the general sites provide custom content downloads and modding community news (mainly about new releases), they rarely host modding tutorials or other custom content know-how. This being the case, *GTA* modding knowledge is shared almost entirely via discussion forums. In addition to the continuous question/answer type of information sharing, step-by-step tutorials and technical documentation is also occasionally posted to the forums. These helpful posts will usually be “pinned” by the moderators, and remain easily accessible at the top of the topic listings. GTAForums has recently opened a new forum dedicated entirely for posting tutorials.

As is the case with the other modding scenes discussed in this report, the *GTA* scene extends outside the community sites. Many fans have established personal sites where they can present their own creations to the community. Modders often focus primarily on one form of custom content¹¹⁵ - or on

¹¹⁵

<http://www.crappy-mods.cjb.net/>,
<http://gunslinger.codenamenetwork.com/Gunslinger%20Vice%20City.htm>

modding tools¹¹⁶ - and gain reputation in the community with their specialty. Since *GTA* custom content files are rather small (< 1 Mb), distributing them via personal sites is generally not a problem. Nevertheless, the files are usually also uploaded to community site archives. Some sites have the habit of adding custom content files to their archives without the author's consent. This has caused a bit of bad blood in the *GTA* scene, though the author's are usually properly credited.

Community projects

One consequence from the lack of official support is that *GTA* modders have not been able to produce total conversions of the game, at least not to the extent *Half-Life* modders have. Nevertheless, there are some modding projects for the *GTAs* that have amassed quite dedicated communities of their own, though these are not so much fan communities as "working groups": instead of having fans create additional content for a total conversion developed by a mod team, the mods themselves are produced from the ground up by the community.

Two such projects - *GTA: Liberty City* and *Myriad Islands* - currently take place at the *GTAForums*, where dedicated forums have been set up for both. On *GTA: Liberty City* fans work collectively to import the entire game world of *GTA3* to the *Vice City* engine. The point is to allow players to take advantage of *Vice City*'s improved mechanics and traverse the streets of *Liberty City* on motorcycles and helicopters. Furthermore, the port will allow players to use *Vice City* mods in *Liberty City*. This is especially important, since *GTA* modding currently has focused heavily on *Vice City* after the sequels release.

While *GTA: Liberty City* has been primarily a conversion project, *Myriad Islands* is all about new custom content. The project coordinators/initiators provided the community with a huge empty map (replacing *Vice City*), and then invited fans to fill it with their own buildings and other objects. The basic guidelines of the mod are discussed together, but outside that participants are free to build whatever they want on their own lots. Despite this freedom modders still like to discuss concepts beforehand and post work-in-progress images to the forums for evaluation.

¹¹⁶ <http://www.steve-m.com/>, <http://www.turnipfan.com/>



The Myriad Islands community mod is developed via GTAForums, where fans share their works and ideas. The project also has a dedicated IRC channel.

Both community projects on the GTAForums are lead by some of the most experienced *GTA* modders, and the communities collaborating on the projects seem to be particularly tight-knit. Another similarly impressive community mod project is the *Berlin Mod*, where German *GTA* fans collaborate to create a playable Berlin city center for *GTA3*. *Berlin Mod* is also developed via discussion forums¹¹⁷.

The only *GTA* mod with what could be called a fan community of its own is *Multi Theft Auto (MTA)*, the most popular multiplayer mod for the *GTAs*. However, unlike *Half-Life* fans who actively produce and share custom content for their favorite mods, the fans of *MTA* only share gaming experiences, offer suggestions, and report bugs. The simple reason is that *MTA* does not currently support fan-made add-ons. Technically speaking much of the custom content created for the single-player versions could be used in *MTA* also, but the *MTA* authors have prohibited it because it would make cheating much too easy. It is up to the server administrators to enforce the rule.

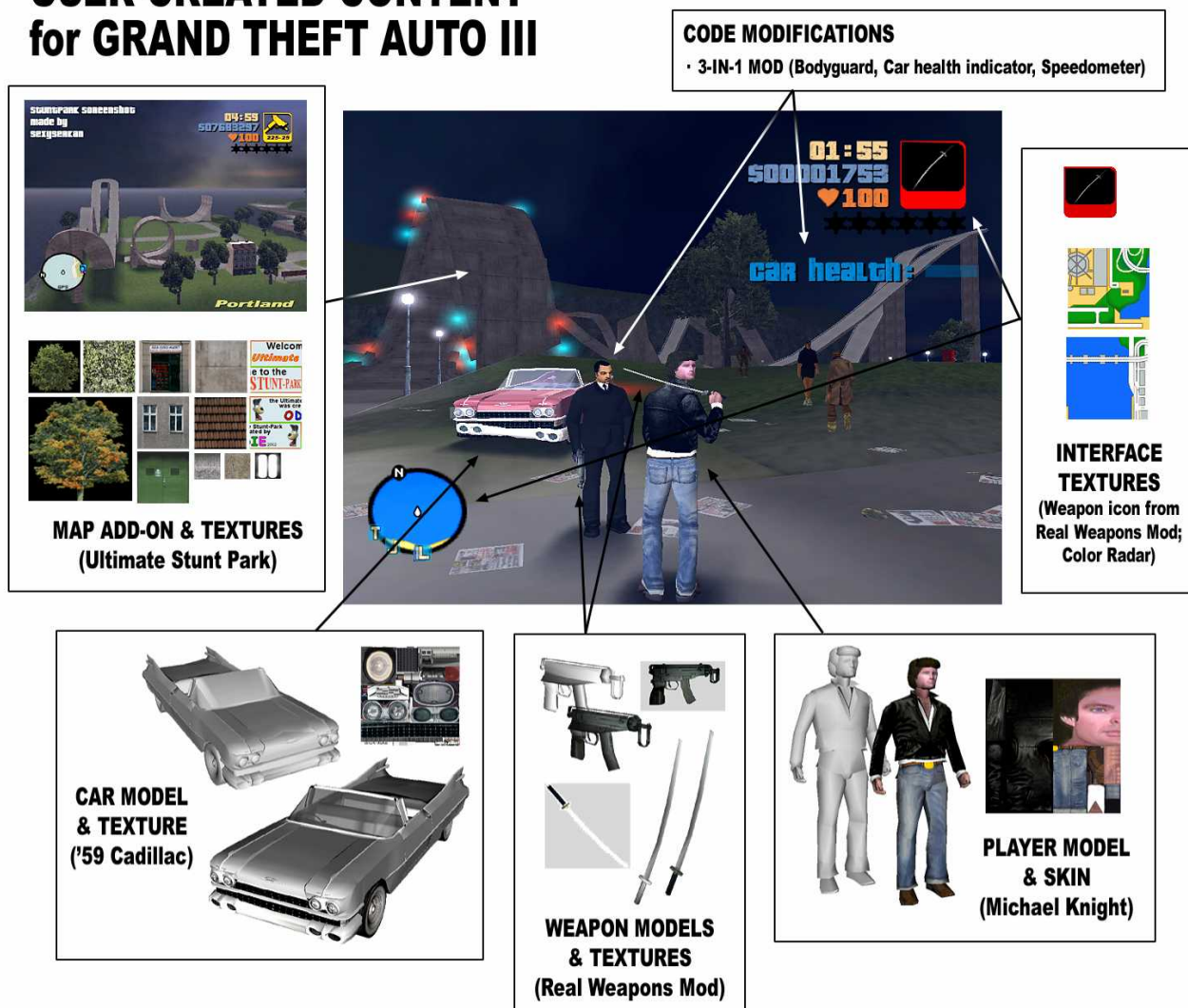
¹¹⁷ <http://forum.gtareactor.de/board.php?boardid=28&sid=>

5.3. User-created content

While extensive total conversion mods are dominant in the *Half-Life* modding scene, and the output of *The Sims* modders consists only of separate small-scale add-ons, *GTA* modders fall somewhere in the middle. Mostly *GTA* modders only create separate modifications: skins, player models, vehicles, weapons, textures, map add-ons and code modifications. However, sometimes dozens of separate modifications from various authors are collected together, and these collections are then referred to as total conversions. Though they can certainly be impressive in scope - some collections include well over hundred different modifications - they are not quite as "total" as usually is understood by the term. The changes they make to the visuals and mechanics are too superficial and random to amount to what could be called a completely new gaming experience. They are more like alternative/improved versions of the original games. The *Myriad Island* community project discussed briefly above is perhaps the closest thing to a total conversion *GTA* modders have yet produced, but even it retains all the mechanics of the original game.

The following dissection illustrates how some of the custom content types come together in *GTA3*. Each custom content item has been installed separately into the game.

USER-CREATED CONTENT for GRAND THEFT AUTO III



5.3.1. Map add-ons

Map add-ons are three-dimensional constructions of varying complexity, which extend the original city maps of the GTAs. They can be assembled either from the games’ original 3d objects, or from custom-made objects created in a modeling utility.

The unique design of the game world was one of the most revolutionary features in *GTA3*. Instead of having a number of separate levels that follow each other as the player advances in the game, the whole Liberty City was one huge continuous playfield. *Vice City* followed the same design, only this time the city also included some playable interiors. This kind of game design naturally means that mapping for the GTAs is very different than mapping for the more traditionally built games

such as *Half-Life*¹¹⁸. Instead of self-contained maps, *GTA* mappers create add-ons to the original city maps. These map add-ons may replace some of the original map objects, and range from new buildings and bridges to complete islands and districts. In many ways the smaller map add-ons resemble large *Half-Life* prefabs, while the most extensive add-ons are comparable to full maps.



GTA map add-ons vary in scope and function. The more extensive add-ons are usually built on new islands over the games' vast empty water areas. From top left: Rockstar Heights, *Vice City* Thunderdome, Il Vecchio Airport, and Miss Liberty.

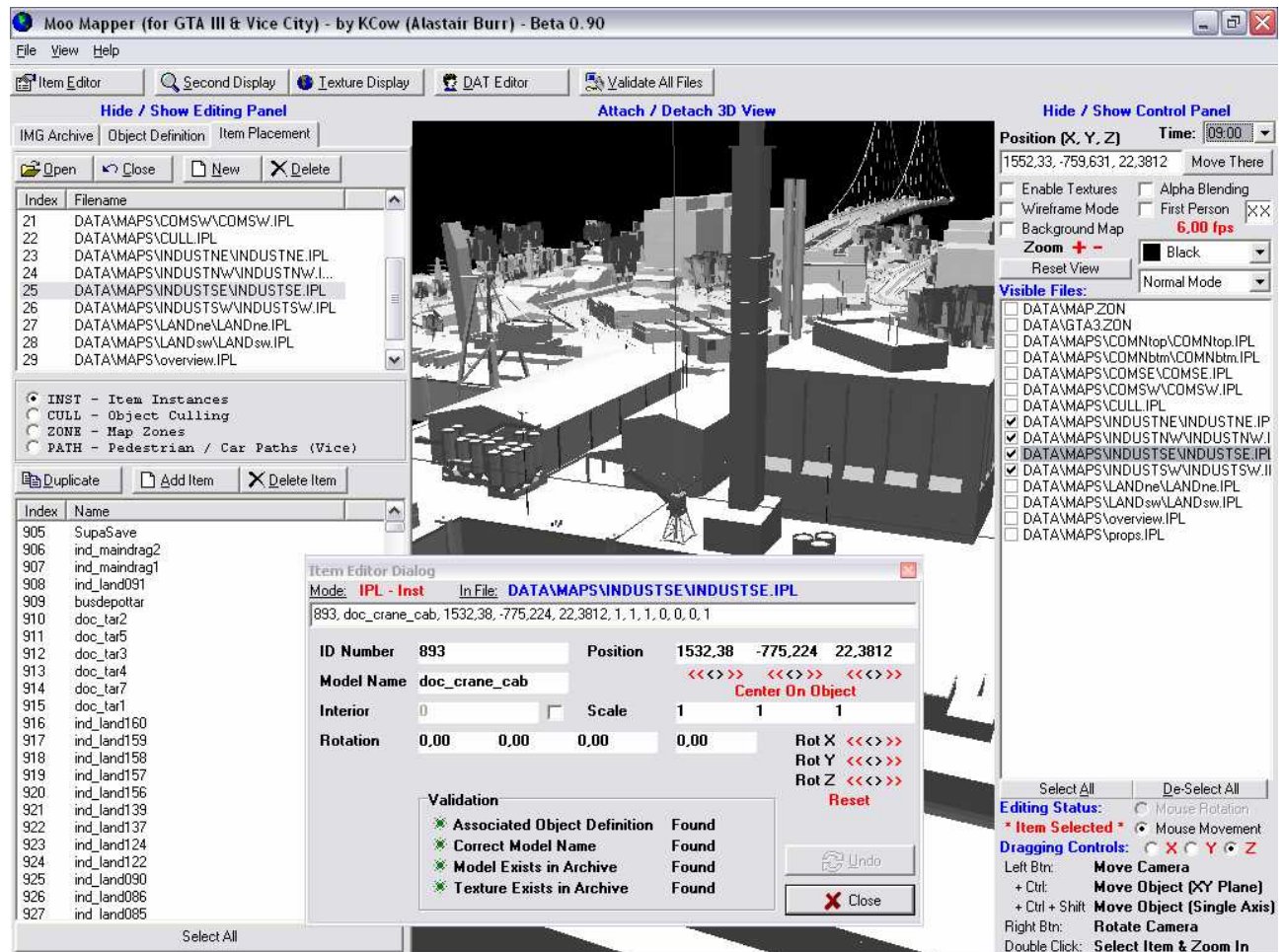
The first widely circulated map add-on for *GTA3* was a bridge that allowed players to drive to an impossible to reach area of the original map.¹¹⁹ Since then modders have steadily gained a better understanding of *GTA* mapping and been able to create increasingly complex add-ons. A major leap forward came with the release of a fan-made map editor called Moo Mapper. While

¹¹⁸ *Half-Life* (as a single-player experience) actually falls somewhere between the level-based games and *GTA3*. The load times between maps are very brief and there is a natural "flow" from one map to another.

¹¹⁹ The area, called "Ghost City", was not meant to be playable. It was only used in the opening cinematic sequence.

the first add-ons were created by manually editing text configuration files, Moo Mapper provided mappers with a graphical interface. Unlike Valve Hammer Editor Moo Mapper itself does not have any content creation tools; it only allows modders to add, remove, and manipulate 3d objects on the *GTA* maps. The objects themselves (i.e. the parts that make up the map add-on) have to be created and textured with a separate 3d application. A popular tool among *GTA* mappers is Zmodeler, a rather crude shareware application, which has built-in support for *GTAs'* (RenderWare) file format. Even mappers who use professional modeling tools need Zmodeler for converting the files. Once created, the custom models still have to be imported into the game data file with a hacking utility such as IMG Tool before they can be accessed within Moo Mapper.

Creating map add-ons does not necessarily involve any modeling, as they can also be constructed entirely from the *GTAs'* original map objects. Sometimes the original objects are retextured for the map add-ons. In addition to the Moo Mapper and (possibly) modeling software, *GTA* mappers usually also need a collision file editor for defining the space custom objects occupy on the maps, and a path editor for defining the paths computer controlled pedestrians and vehicles follow on the customized map area. There are fan-made tools for these purposes also.

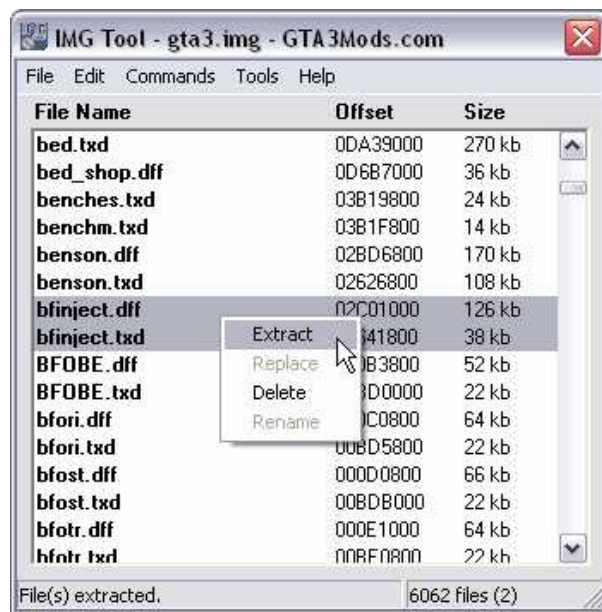


The fan-made Moo Mapper is perhaps not the user-friendliest of applications, but still a huge leap forward from the prior method of editing map configuration files with a text editor. Moo Mapper is currently an open source project.

With proper tools in their hands *GTA* community has been able to create more and more complex map add-ons. Two distinct themes emerge from the map modifications: fun and function, with the former being much more popular. Especially popular are stunt parks/islands where ramps, half-pipes, roller-coaster type tracks and other such structures provide perfect opportunities for extreme car stunting. Some other add-ons, on the other hand, serve a specific purpose in regards of the gameplay, usually providing easier access from place A to B. An impressive example of a functional add-on is The Intercity Highway, which connects all three islands of Liberty City with a huge sprawling highway.¹²⁰ *GTA* mappers have also recreated some real-world add-ons, such as the Statue of Liberty and the Eiffel Tower, but these are quite rare. There is, however, an extensive mapping project called Berlin Mod underway which attempts to create an approximation of the whole downtown of

¹²⁰ At the beginning of *GTA3* only one of the three islands that comprise Liberty City is playable. The bridges and underwater tunnels to the other islands are opened once the player has completed several missions. It can take up to 20 hours of playtime to open the third island.

Berlin and its famous landmarks in *GTA* style.



IMG Tool is a simple hacking utility, which allows modders to import and export texture and model files from the *GTAs'* game data files. It is an essential tool for all modders, but also for anyone wishing to use the mods.

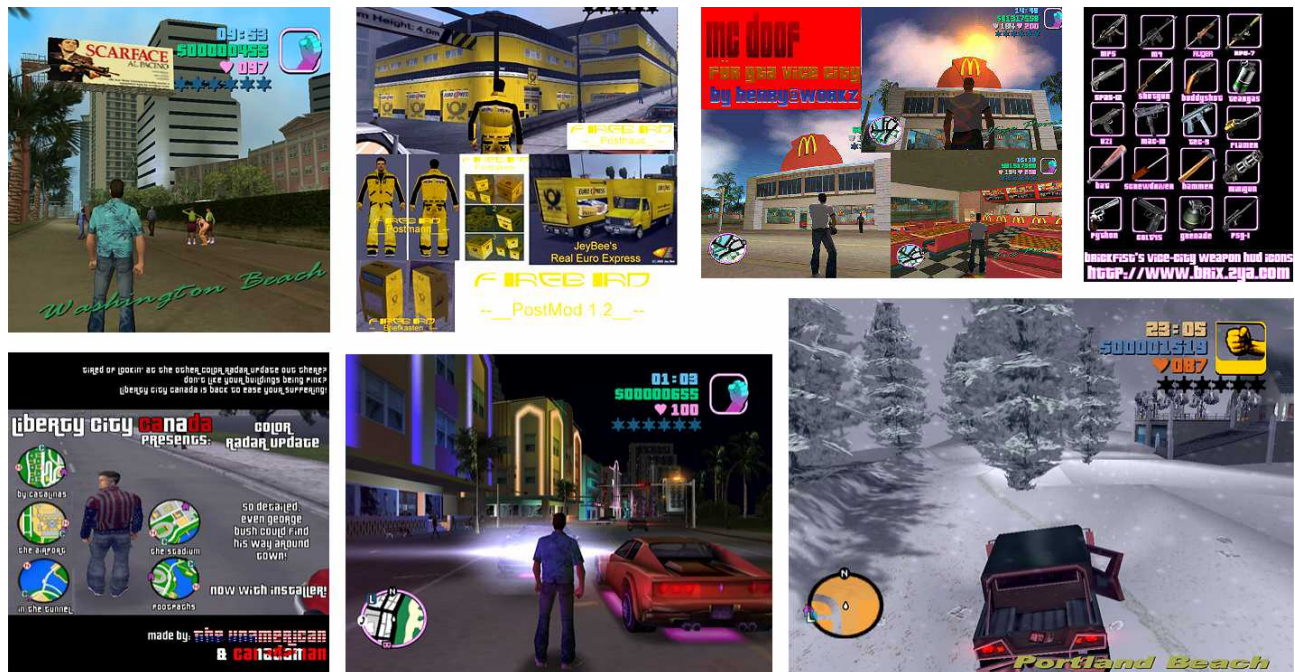
The installation of map add-ons - like all *GTA* modifications except for skins - can be rather complicated and error-prone. The complexity depends on the amount of custom models and textures used. The user must manually edit configuration files and use a hacking utility such as IMG Tool to import custom models and textures into game data files. Fortunately some of the most extensive add-ons come with automatic installers. Another potential problem is that since the add-ons are placed on the same city map, they might overlap each other causing complications.

5.3.2. Textures

Textures are 2d bitmap images used in various ways to customize the *GTA* game world. Unlike in the *Half-Life* modding scene, where textures are only raw material for custom mappers, in the *GTA* scene fan-made textures and texture sets are also commonly used as stand-alone modifications.

Most of the separately distributed (i.e. those not attached to map add-ons) custom textures are designed to replace one or many of the games' fictional storefronts and roadside billboards with real-world brands and advertisements. These replacements range from simple signboards to more extensive sets that change the appearance of a building completely, or replace several original textures from various places around the game map. Some sets also include custom skins and textures for vehicles. In addition to the real world "branding" textures, *GTA*

modders have created environmental textures¹²¹ to customize the games' sky, water, pavement etc. One of the most impressive environmental texture sets is The Snow Mod (Die Schneemod), which gives the streets of Liberty City a coating of snow. The set also replaces raindrops with snowflakes and includes configuration files that make the streets slippery and nights last longer.



GTA fans create textures for various customizing purposes. From top left: Scarface ad from Hot Billboards collection, EuroExpress texture set (incl. player skin), "McDoof" restaurant replacement set, weapon hud icons, Color Radar, under chassis neon glow effects, and The Snow Mod.

In addition to the maps, GTA modders also use textures to customize other aspects of the games. Effect textures are somewhat similar to sprites in *Half-Life* modding. Instead of being applied to map surfaces, they are used for such purposes as changing the colors of car headlights, making the blood look more realistic, giving bullets Matrix-style ring trails, and adding neon lights underneath cars. Interface textures also resemble sprites. Most common interface textures are customized radars and weapon icons, but fans have also created replacements for the GTAs' load screens and display fonts.

Regardless of their usage, the texture images can be created in any graphics software. Additional fan-made modding tools are needed for exporting and importing the texture files, and for compiling the bitmaps into GTAs' texture format. (Sometimes

¹²¹ The texture categories discussed here are made up only for the purposes of this presentation. They are not something that GTA modders themselves use. On download sites all textures and texture sets are often clumped under the ubiquitous category of "mods".

the textures are distributed in generic bitmap format, leaving the conversion task to the user.)

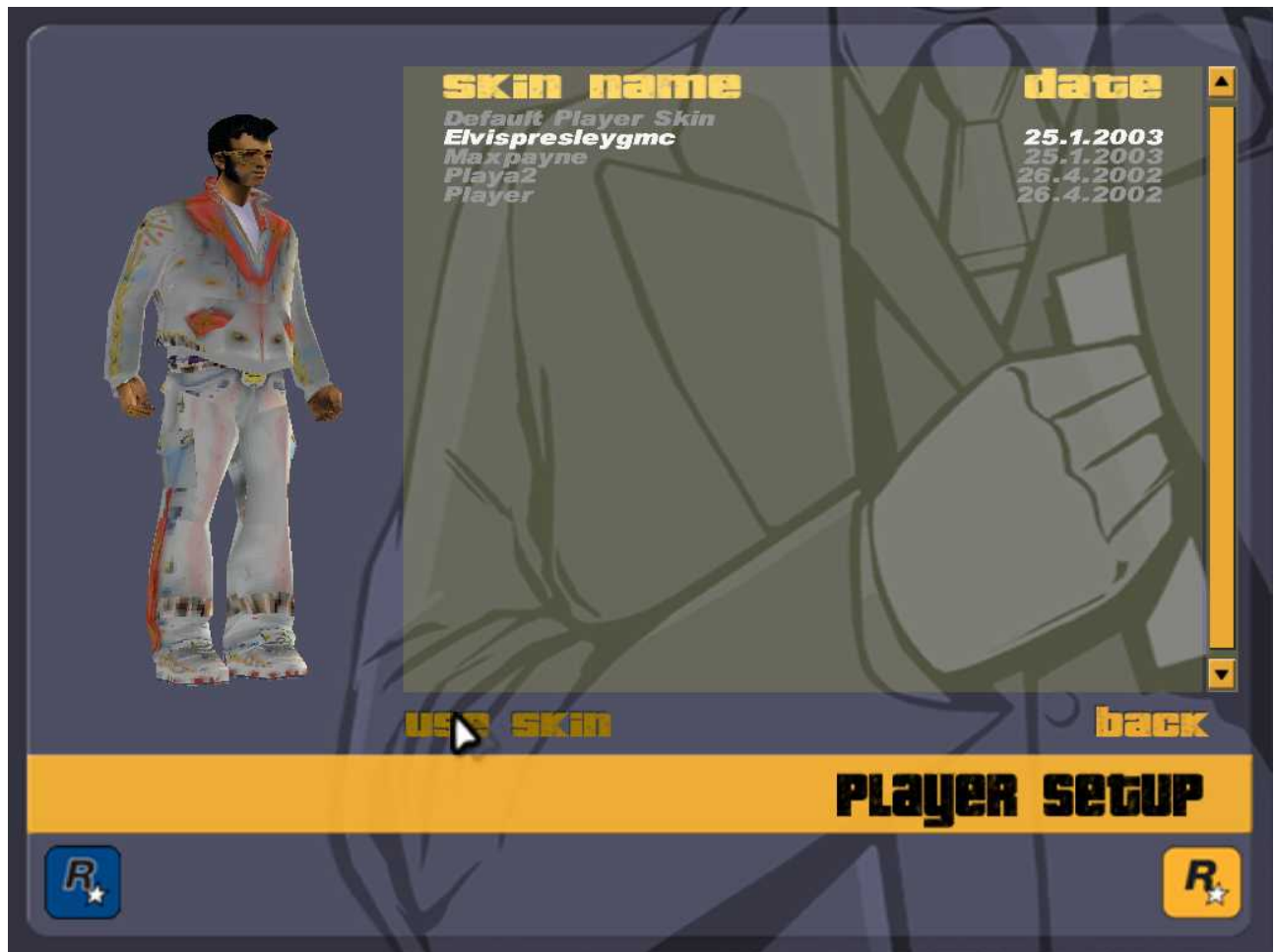


TXD workshop is one of the fan-made utilities that can compile bitmap images into *GTAs* proprietary texture files. It can also export original textures to bitmaps, so that they can be used as a basis for custom textures.

5.3.3. Player models and skins

Like in all 3d games, the characters in *GTAs* are combinations of three-dimensional meshes, which define the character's physique, and bitmap skins, which wrap around the mesh to define character's facial features and clothing.

Customizing the player character with skins is the only officially supported form of *GTA* modding. To Rockstar's credit it must be said that creating and installing custom skins could not be much simpler. Since the skins use the generic BMP format, any graphics application will suffice. Once created and saved into the skins folder, custom skins can be applied to the player character from an in-game menu.



Customizing the player character with skins is officially supported in both GTAs. Installed skins can be previewed and selected from an in-game menu. The player skin can be changed even in the middle of the game.

Unsurprisingly, skinning is very popular in the *GTA* modding scene, with some sites offering hundreds of custom skins for download. The level of customization varies from minor tweaks (e.g. adding facial hair, tattoos, or sports team logos to the original skin) to complete makeovers. In the latter category recreations of popular culture characters are common, but many skins are also made just to provide the player character some nifty new outfits. Skins are sometimes distributed in thematic collections, such as the Armani suit set and the Hawaiian shirt set for *Vice City*.

The physique of the GTAs' player models sets some limitations to skin makers. The anonymous player character in *GTA3*, for example, has baggy pants, bulky leather jacket and a pompadour. These features are sculpted into the 3d model and cannot be disguised with skins. In *Vice City* there are several player models (the player can change clothes in the game), but only the default outfit (jeans with a Hawaiian shirt) can be customized with the "official" skinning method.

Unofficial hacking utilities allow skin makers much more flexibility. The default player model can be replaced with any of the non-player character models (e.g. one of the females), after

which custom skins can be created according to the NPC's body type. Of course, *GTA* gamers who wish to use these skins must also know how to use the hacking utility to switch the models. Instructions are usually found in the readme-files. With hacking tools it is also possible to replace the skins of non-player characters. Nude skins for the games' female characters are particularly popular downloads. Unlike player skins, the NPC skins use *GTAs'* proprietary texture format, which means that the skin maker also needs texture conversion tools.



GTA custom skins and models (from top left): Elvis and Dead Tommy are regular player skins. Annie Oakley, Bride (from *Kill Bill* movies) and Lara Croft require the player model to be first switched with a female NPC model (also distributed with the skins). "Big Head", Bender, Gohan and Cassandra are custom models for *GTA3*. Super Mario is one of the first custom models for *Vice City*.

While there is no shortage of custom skins, custom character models are much more rare. Furthermore, practically all the available custom models are for *GTA3*. The RenderWare engine was updated for *Vice City*, and the unofficial import/export plug-ins modders used to convert the *GTA3* models into editable format turned out to be incompatible with the new version¹²². Only recently, as a result of intensive collaboration via *GTAForums*, did the modders come up with working plug-ins for the sequel¹²³. There are, however, still many things about the new file format modders have not figured out. Character animations, for example, cannot be edited yet. (The animations of *GTA3* characters cannot be edited either - they are hard-coded into the game executable - but they "scale" to models with different proportions.)

Most of the custom models available for *GTA3* are recreations of popular culture characters, such as Solid Snake from *Metal*

¹²² Unlike *GTA3* characters, *Vice City* characters use skeletal animation.

¹²³ One frustrated modder at the *GTAForums* offered 100 dollars/euros to whom ever who would come up with the solution.

Gear Solid games (Konami 1998), T-101 robot from the movie *Terminator* (1984), and Bender the robot from TV's *Futurama* (USA 1999-2003). Custom models are almost always accompanied with custom skins. Although many download sites also offer custom models for *Vice City*, these are actually re-skinned NPC models from the game. As mentioned before, with hacking tools the player model can be replaced with any of the NPC models. Since the import/export solution for *Vice City* was found this spring, a few "real" custom models have been released. One of the first ones was Nintendo's mascot Mario (exported and converted from N64 console game).

5.3.4. Vehicle models and textures

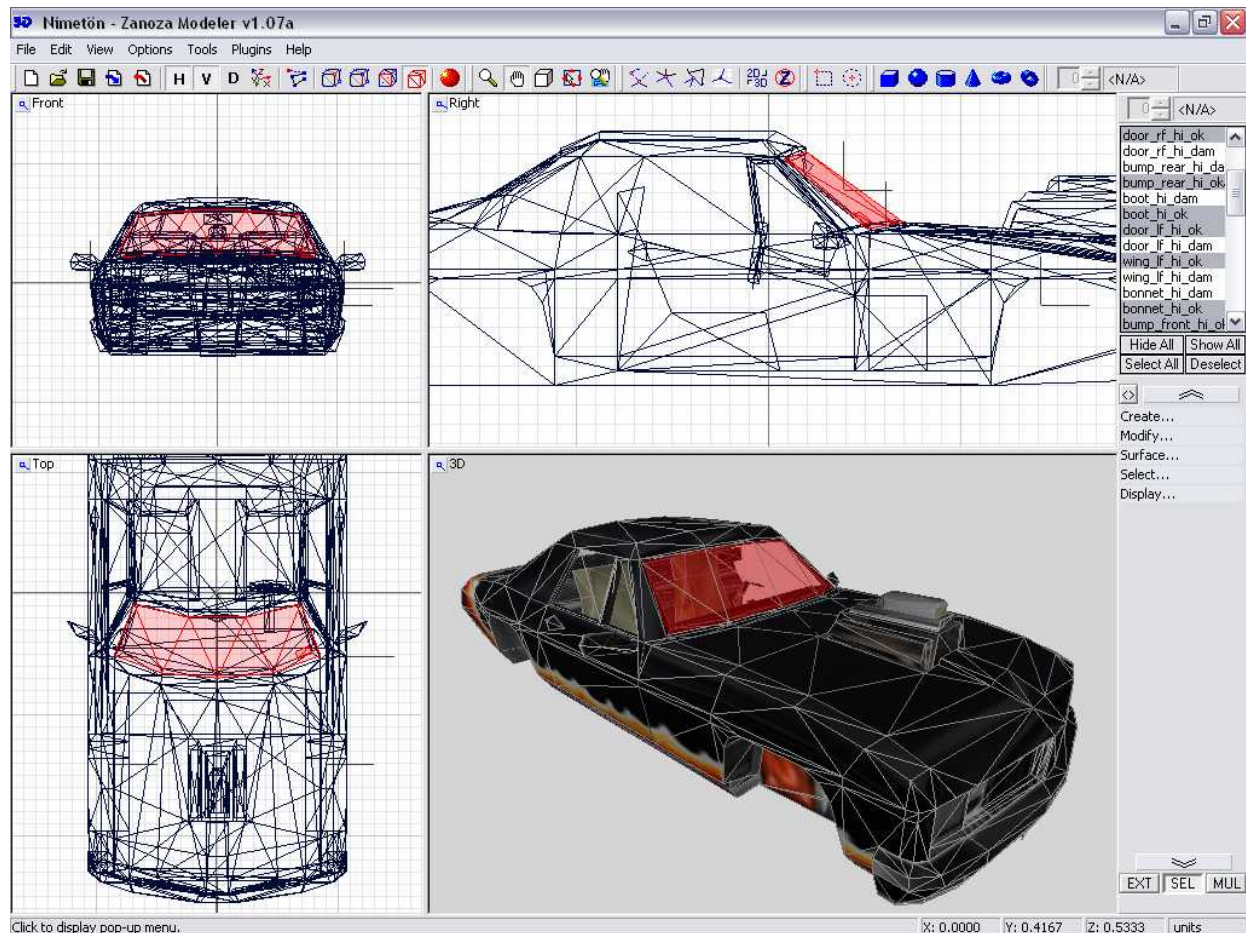
Vehicle models for *GTA3* can be either cars or boats, while *Vice City* also has motorcycles and helicopters. The color of the vehicles is defined in configuration files. Textures are used for details, such as the front mask, wheels, interiors and possible logos/paintings.

While the *GTA* character models may be rare, there is no shortage of new vehicles, especially cars. The file archive of a German mod community site *GTA3.net*, for example, lists over 700 user-created cars. Most of the custom cars are recreations of real-world cars to replace the fictional vehicles of the original games. The *GTAs* have a wide variety of cars to replace, which also reflects in the diversity of fan community's output. Besides supercars, such as Ferraris, Porsches, and Lamborghinis, fans have also produced plenty of more mundane replacements, such as Fords, Minis and Volkswagens, as well as specialty vehicles such as Formula 1s, tractors, monster trucks, Batmobiles and go-karts. For the *Vice City* fans have also modeled quite a few motorcycles, and some helicopters and boats, but overall these represent a small minority compared to custom cars.



An assortment of custom vehicles for the GTAs. Fan-made vehicles are often more detailed than the originals, which can cause slowdowns, especially if several custom cars are installed simultaneously.

Vehicle models are often converted from other racing games (the aforementioned Zmodeler can convert files from various games), but many are also created from the scratch. Car modeling is, after all, perhaps the most popular area of interest among 3d hobbyists. The tools involved in vehicle modeling are similar to character modeling. Since vehicle models obviously do not have character animations, the same import/export plug-ins are compatible with both *GTAs*. If the vehicles are created from scratch, or if the customized model differs substantially from the original one it replaces, a tool such as Collision File Editor is needed to define the space that the model occupies in the game world.



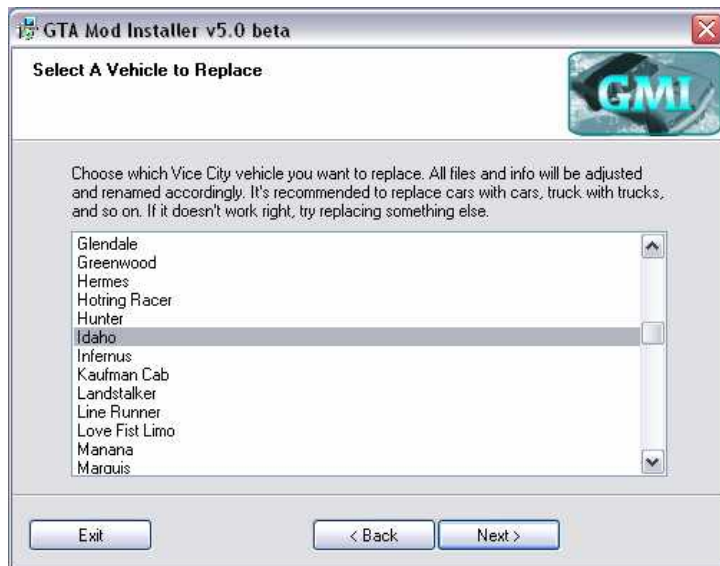
Though crude compared to professional modeling tools, Zmodeler is very popular among *GTA* modders because it is free and has built-in support for the *GTAs* proprietary file format. It also supports various other games, making vehicle conversions easy.

Although the overall coloring of vehicles is defined with RGB values in a configuration file, custom textures can be used to add various details to the car body and interiors. Fans have also created custom textures for the *GTAs*' original vehicles, as well as wheel texture sets with photographs of luxurious real-world rims to replace the original ones.

In addition to the colors, the handling and the wheel size of the cars is also defined in configuration files. (For the vehicles that use custom-sized wheels, a fan-made patch called WheelMod has to be installed.) While the configuration files can be edited with any text editor, there are also simple utilities for the task. Customized configuration files are usually not distributed with the models; only the lines that need to be added/changed are included in the readme-file.

Installing custom vehicles can be complicated for the average gamer (even though the models are usually accompanied by readme-files with step-by-step instructions). Not only must the player hack the game data files to import the models into the game, he also has to edit various obscure configuration text files. Recognizing this issue, resourceful modders have come up with tools such as the *GTA* Mod Installer that automate the installation process to a great extent. An additional problem is

caused by the fact that custom cars always replace one of the original cars. It is inevitable that some of the downloaded custom cars are designed to replace the same car. Only one of them can be installed at a time. *GTA Mod Installer* allow players to select which original vehicle is replaced, but selecting a different one that the author has intended can cause all kinds of problems.



Mod Installer greatly simplifies the rather complicated installation process of custom vehicles. It also makes back-ups automatically.

5.3.5. Weapon models

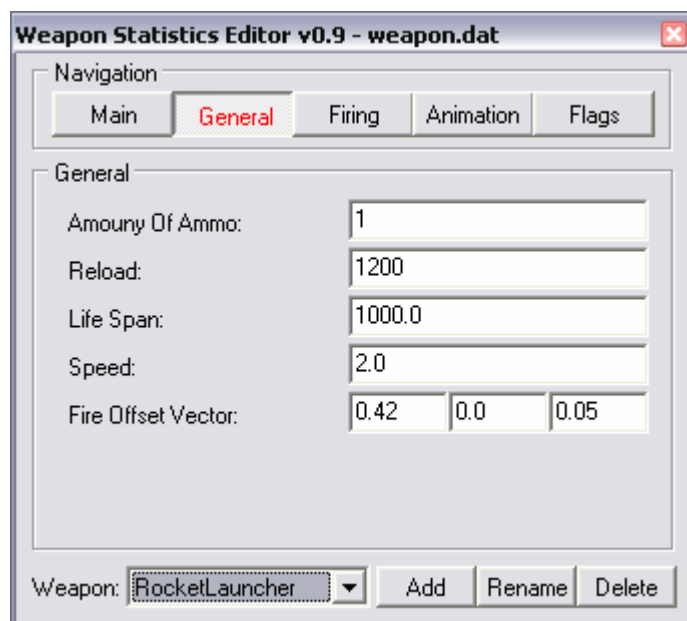
GTA weapon models are technically similar to *Half-Life* weapons, but unlike in *Half-Life*, there is only one model for each weapon in *GTA*. Weapons are also textured with just one bitmap skin, limiting the amount of detail possible (but also making (re-)texturing much easier).

Considering the games' theme, it is perhaps surprising that weapon modeling is not particularly popular among *GTA* modders. Perhaps this is because the weapons are not so prominently on display on a third-person game such as the *GTAs*. Overwhelming majority of the custom weapons *GTA* fans have produced are recreations of real life weapons, but there are also some fictional or tongue-in-cheek replacements, such as water cannons, toy hammers, and lightsabers. Quite a few weapons are converted directly from other games (e.g. *Counter-Strike*, *Max Payne*). Custom weapons are often distributed in sets.



Custom weapons for the GTAs (from top left): Zaku machine gun, Beretta 92, Max Payne weapon set (converted from the game), Uzi, Real Weapons Mod, and SuperSoaker (replaces flamethrower).

The techniques and tools used in weapon modeling are pretty much identical to car and character modeling. The installation procedure is also similar and requires unofficial hacking tools. Like vehicles, weapons have attributes (ammo capacity, firing rate, damage inflicted, etc.) that are defined in configuration files. These can be edited manually or with specialized utilities. Customized configuration files for the GTAs' original weapons are also available for download.



Though Weapons Statistics Editor is mainly used for changing the attributes of the GTAs' original weapons, with its graphical interface it also makes it easier for modders to define the attributes of their custom weapons.

5.3.6. Code modifications

Code modifications are modifications made into the *GTAs*' main script file¹²⁴. They can either give the player new abilities, or add new functionality or new missions to the game.

It's important to notice that *GTA* modders do not have access to the source code like *Half-Life* modders do. Instead of being programmed with C++ or other real programming language, code modifications for *GTAs* are written in a proprietary scripting format. This obviously limits the flexibility allowed for modders and consequently the scope of modifications. *GTA* coders cannot create entirely new game mechanics, they can only tweak and reassemble the existing mechanics in new ways. The same applies to fan-made missions, whose objectives are always derivatives of the original missions.

That said, within the given limitations *GTA* community has been able to produce a quite diverse and innovative array of code modifications to spice up the games. An example of a simple yet useful code modification is a mod that adds a speedometer and a car damage level indicator to the game interface. This kind of modification would not be possible if the original game did not already have mechanics to measure car speed and damage. What the modification does is make this information visible to the player.

Many of the small code modifications are more or less cheats, as their primary purpose is to make the gameplay easier: give the player super powers, pair him with a bodyguard, make him invincible, make the cars indestructible, make the best cars/weapons easily available, allow saving at any time, etc. Some code mods are just silly; one for example fills the *Vice City* streets with drunken (i.e. recklessly speeding and swerving) drivers. The lasting appeal of these modifications may be limited, but they can nevertheless be refreshing for those who have already played the games through over and over again.

¹²⁴ In the *GTA* community the word "mod" often refers specifically to code modifications.



Examples of code modifications (from top left): “Car Spawn” allows player to spawn any vehicle at any time. “Bodyguard” provides the player with a gang of protectors. “Backseat” adds a first-person camera angle. “Neo” adds a number of Matrix-influenced features, including the ability to fly.

There are also more extensive code modifications, which make several, often somehow related changes to the game. One such mod, called *Neo*, includes all kinds of Matrix-inspired features, such as ability to fly and run through buildings, and of course the famous “bullet time” slow motion. Another one called *State of Emergency* turns the whole *Vice City* into a war zone, filling the streets with an aggressive army. The player can hijack tanks and assault helicopters from all over town, and even recruit a mercenary army of his own. Perhaps the most extensive code modification available is the *New Vice City*. Just to mention a few of the dozens of changes: damage and speedometers are added to the interface, new camera angles are available, cars need to be filled up, the player can call up a taxi at any point, missions are delivered via cell phone, the game time is slowed down closer to real world time, player must escape from jail after being arrested, characters from completed missions can be recruited as sidekicks, the whole police force can be “bought”, after which it will not harass the player anymore, and so on.

New Vice City also includes some new missions (as well as removes some “shitti ones”). While code modding is often referred to as mission scripting in the *GTA* modding community

(missions are defined in the main script file), completely new fan-made missions are actually quite rare. More common is to somehow alter the existing ones: slow down or remove the mission timer, make better cars/weapons available, make the rewards bigger, etc. The rarity of completely new missions is mostly due to the complexity of the *GTAs'* scripting format. This is also why the few new fan-made missions tend to have simple racing or collecting objectives. It should be noted, though, that *GTA* modders are still constantly learning new things about the scripting format and have been gradually able to create more and more complex missions. The lack of developer support and documentation naturally hinders the progress.

```

:Label04FE09
00D6: if 0
80E0: NOT player $PLAYER_CHAR driving
004D: jump_if_false $Label04FE2B
0001: wait 0 ms
0050: gosub $Label05008C
0002: jump $Label04FE09

:Label04FE2B
03D5: remove_text "GEN1_20" ;; ~g~Get into a vehicle.
00BB: text_lowpriority "GEN1_21" 5000 ms 1 ;; ~g~Get to the~h~ Pay 'N' Spray~g~ in~h~ Vice Poi
0335: set_free_payhspray_to 1 (true)
018A: $7E2 = create_checkpoint_at 327.2 429.9 10.3

:Label04FE5C
0001: wait 0 ms
00D6: if 21
0329: garage $100 respray_done
0329: garage $101 respray_done
004D: jump_if_false $Label04FE97
0164: disable_marker $7E2
0001: wait 4000 ms
03D5: remove_text "GEN1_22" ;; Drive your vehicle into the spray shop to lose your ~h~wanted lev
0002: jump $Label04FF43
0002: jump $Label04FEB0

:Label04FE97
00D6: if 0
810F: NOT player $PLAYER_CHAR wanted_level > 0
004D: jump_if_false $Label04FEB0
0002: jump $Label04FF43

:Label04FEB0
00D6: if 0

```

Barton Waterduck's Mission Builder is one of the two popular tools for editing *GTAs* mission script files. The competing tool takes a little different approach to mission scripting, which has lead to a division among *GTA* coders. Animosity is not uncommon between the two camps.

All code modifications, whether mechanics tweaks or new missions, are coded into the games' main script file. Obviously there are no official editors available for the task. Most *GTA* modders use a fan-made tool called Mission Builder (separate

versions for both *GTAs*)¹²⁵. Mission Builder is a spiced-up text editor that can decompile and compile *GTAs*' proprietary script file format. Some extra features, such as macros and bookmarks help the workflow, but the lack of a debugger means everything is built more or less by trial and error. The author of the tool has provided a very thorough tutorial.

Code modifications can usually be installed simply by replacing the original script file with the modified one. There are, however, some annoying issues to deal with. Perhaps the most annoying thing is that a new game must be started from the beginning each time a code modification is installed, even if it is just a simple mod such as the speedometer. It is also impossible to have more than one code mod installed simultaneously (unless the user himself knows how to edit the script).

5.3.7. Major modifications

As has been already discussed, *GTA* modders have not been able to create total conversions for the games. It might well be that so much of the games' inner-workings have been hard-coded that total conversions comparable to *Half-Life* mods will never be possible. However, within the given limitations, *GTA* community has still managed to produce modifications that can certainly be called major. These major modifications could be loosely categorized into mod collections, community projects and multiplayer mods.

Mod collections combine numerous custom content items gathered from the community into a single installable file. The emphasis is always on custom cars and textures, but most collections also include few map add-ons, some weapon replacements, and even simple code modifications. All in all some of the most extensive collections may include over a hundred individual modifications. The collections are put together by individual modders who often also include some of their own works in the package. Putting together a diverse collection is by no means a trivial task. Installing even a single custom content item to *GTAs* can be laborious, and installing two or more simultaneously often leads to all sorts of complications. Creating a collection that automatically installs dozens of them is obviously quite an undertaking and requires some planning. Securing permissions from all the modders whose works are to be included can also be a tedious process.

Of the handful of mod collections that have been released so far, perhaps the most popular is *RealGTA* for *GTA3*. As the name suggests, the overriding theme in the collection is realism: the

¹²⁵ Mission Builder's compiler/decompiler was the result of intensive collaborative reverse engineering of *GTA3*'s script format via *GTAForums* and private IRC channels.

GTA3's fictional cars are replaced with real models (Ferraris, Porches, BMWs, etc.), store fronts with real world franchises (McDonald's, Shell, IKEA etc.), billboards with real advertisements (Coca-Cola, Pepsi, AMD etc.), and weapons with real makes and models. There are also some new textures and interface elements, as well as a Liberty Island map add-on, complete with an accessible Statue of Liberty. Some of the new car models in *RealGTA* are impressively detailed, but this is not without its drawbacks: the mod requires a much more powerful computer than the original game.



The author of the popular *RealGTA* mod has collected dozens of custom vehicles and textures from the community and compiled them into a single installable file. The original authors are of course credited in the accompanying readme-file.

Another popular collection, known as *KillerKip's Mod Pack*, is more of the patchwork variation. While most of the vehicles are replaced with real world models, there are also many not so real elements, such as various *Star Wars* (1977) aircrafts. Other modifications in the collection are equally random: map add-ons include buildings from Terry Pratchett's *Discworld* fantasy novels, currency is changed to euros, dancers in the night club are topless, hidden packages are changed to lollipops, and so on. A small nod to the community are billboards and store signs

that incorporate the nicknames of other fans (e.g. a nightclub sign that reads "MetalDennis' Heavy Liquor"). Of course KillerKip's own insignia can be found all over town.

The community projects - *Myriad Islands* and *Berlin Mod* - that were already discussed earlier are in essence also mod collections. They too combine the separate creations of several modders into a larger whole. What makes them unique is the collaborative working process. Instead of an individual modder gathering published mods from the web archives, the whole community works in the project simultaneously, collaborating via discussion forums. Naturally modders can (and do) also bring their earlier creations to the projects, if the community deems them suitable.

Of all *GTA* modifications the *Myriad Island* project is definitely the most extensive, and the closest thing to a total conversion. The whole gameplay environment from the street signs up is fan-made. The project initiators created an empty map consisting of two islands. The map was then divided into 99 sections and all modders invited to reserve their own piece of land. General themes (industrial, downtown, suburb) were given for different parts of the map, but other than that, modders have been free to build whatever they want on their own lots. Some have built a simple house, while others have constructed a massive complex of skyscrapers. Since there is no single installation file (yet), everyone is free to decide for herself, which add-ons are worth installing. Gameplay-wise there is not yet much to do except to explore the fan-made city, but the coders of the community are developing completely new missions for the mod. Some have already been released. The quality of the custom missions will be a crucial factor in determining whether *Myriad Island* will generate much interest outside the developing community itself. On the other hand the initial idea behind the project was first and foremost to provide modders with a canvas for creation, not so much to create an exciting gaming experience.



The initiators of the Myriad Island project provided the *GTA* community with an empty map and invited modders to fill it with their own creations. All the fan-made map add-ons have to be installed separately. Though a bit cumbersome, this modular structure allows players to install only the add-ons they want. The addictive duck shoot (bottom right) is one of the few custom missions created for the mod so far.

The final category of major modifications consists of two rivaling multiplayer modifications. When the PC versions of the *GTAs* were announced, the biggest disappointment for most fans was the lack of multiplayer capabilities. This is understandable, since almost all new PC releases these days have online features, and the diverse game world of *GTAs* seems to offer particularly rich opportunities for social gaming: car/motorcycle/helicopter races, deathmatches, destruction derbies, gang wars, co-operative play, etc. Ever since the PC version of *GTA3* first came out, fans have been trying to patch the shortcoming themselves.

To certain extent fans have succeeded. There are currently two working multiplayer mods available: *Multi Theft Auto* and *gtaTournament*. Of these two, *Multi Theft Auto*, originated by a Dutch teenager but these days developed by an international team of modders, is by far the more popular one. *MTA* has two gameplay modes: stunt and deathmatch. In stunt mode the player's objective is to do impressive car stunts and earn money. In deathmatch mode the player first chooses one of the available characters (e.g. cop, robber, Mexican) and then starts killing fellow players. Team deathmatches are also possible.

Gangs are formed and gang wars organized at *MTA's* message boards. A passenger function in *MTA* enables gang members to roam around the city in the same car causing all kinds of havoc. Both gameplay modes require dedicated servers. Unsurprisingly, the deathmatch servers are much more popular than stunt servers.



With some clever programming the MultiTheftAuto team has been able to add multiplayer capabilities to both *GTAs*. Although the mod is still too unstable for casual gamers, it has slowly but steadily built a dedicated fan base. The latest version is modestly numbered 0.4.

Overall *MTA* is still a marginal sport, however, and even populated deathmatch servers can be hard to find. Much bug fixing and further development is still needed before *MTA* is able to vow casual gamers. Since the *GTAs* were not designed for online play, some issues may turn out to be impossible to resolve – even if, as rumored, Rockstar is giving the *MTA* team some technical support.

MTAs rival, *gtaTournament* was the first working multiplayer mod for *Vice City*. Since the release of *MTAs Vice City* version, the interest in *gtaT* has been on a decline, however, and these days it can be impossible to find any servers that run the mod. Perhaps the next version, which is supposed to be technically superior to *MTA*, will breathe new life to the mod and its withering fan community. Although many have doubts whether

the announced features can actually be implemented or if the next version will ever even be released.

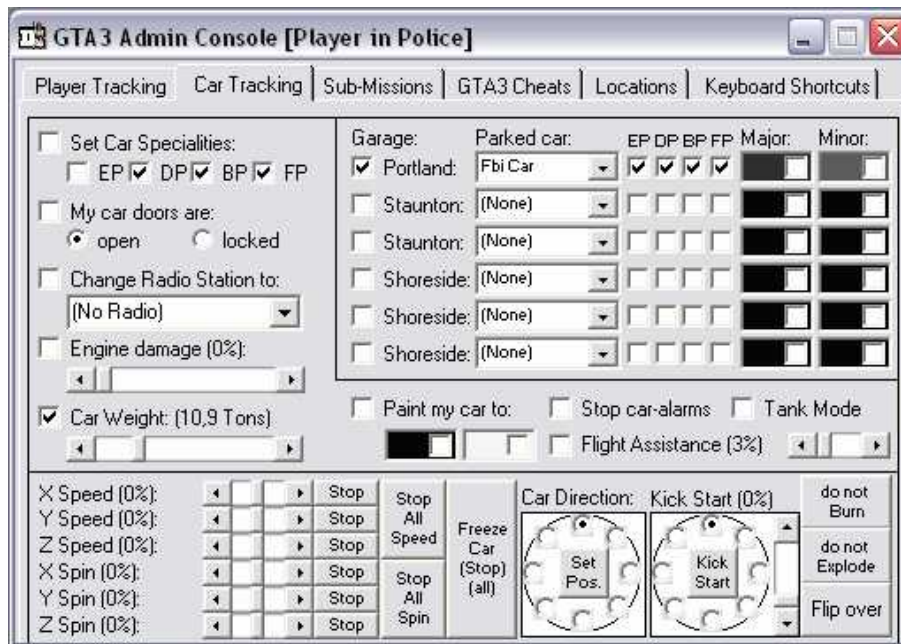
5.3.8. Modifiers

Modifiers (author's term) fall somewhere between mods and modding tools. They are small utilities that allow players themselves to make various adjustments to the game world: enable cheats, change the attributes of cars, weapons or NPCs, jump to any position in the game map, change the weather, etc. Some modifiers have only one specific function, while others enable tweaking in several areas.

Of the single-purpose modifiers one of the most innovative is certainly the Weather Tool. The application connects to an Internet weather service and updates *Vice City's* weather to match the player's local weather (or whichever city the player has selected). The tool also synchronizes *Vice City's* in-game clock to real world time. Another popular modifier with focused functionality is Mess With Those Peds, which allows players to adjust the behavior of the NPCs (aggressiveness, fearfulness, strength, etc.), and determine the distribution of different pedestrian types in any location. With MWTP it is also possible to swap any two characters: policemen with prostitute, gang members with old bag ladies, and so on.

The most versatile modifier is undoubtedly MultiEdit for the *GTA3*. With MultiEdit players can change every imaginable attribute of the vehicles (color, speed, availability, etc.), weapons (damage, ammo, rate of fire, etc.), and characters (aggressiveness, location, density, etc.), as well as tweak some particle effects (explosions, smoke, etc.) and other game world variables. The program seems to have impressed even Rockstar, who took advantage of the tool while building *Vice City*.¹²⁶

¹²⁶ Rockstar has not openly admitted this, but MultiEdit's signature tag has been found on one of *GTA:VC's* stat files.



GTA 3 Admin Console is a popular multipurpose modifier, which allows players to set an exhausting array of game attributes. The tool has to be loaded while the game is running, and attributes can be changed at any time.

5.4. Summary and notes

The driving-action game *Grand Theft Auto III* was released in 2001 for PlayStation 2 console. The game's unprecedented scope and open-ended variety of gameplay earned high praise from the gaming community, while the controversial gangster theme caused disquiet among parents and various interest groups. Despite the controversy (and partly because of it) *GTA3* became one of the best-selling video games of all-time. The Miami Vice inspired sequel *GTA: Vice City* followed a year after to similar success (and controversy). Though the basic architecture of the game remained the same, *Vice City* expanded the play-field even further. Both games were released for PC about six months after their PS2 launch.

Once the PC version of *GTA3* was available, the fans, pooling their knowledge via Internet forums, soon figured out how to modify the game. However, since the game utilized licensed technology, the developer Rockstar could not offer any official support or tools. Nevertheless, by the time the PC version of *Vice City* was released, the community already had fan-produced tools and knowledge available to modify almost every aspect of the game. The same tools and knowledge could generally be applied with minor tweaks to the sequel as well. New tools are developed continuously as the community learns more about the games' architecture.

While the achievements of the *GTA* community are impressive, the fans have not been able to totally overcome the lack of official support. The games were not designed to be extensively modified, and as the modding projects have become

more ambitious, fans have run into increasingly frustrating technical hurdles. The usage of user-created content is also cumbersome and requires specialized “hacking” utilities, potentially alienating more casual gamers.

Rockstar has not setup an official community site for the fans. Instead, the most popular meeting place for the *GTA* fans, especially modders, is the unofficial GTAForums community. In addition to GTAForums there are dozens of other general fan communities, many of which host custom content downloads. Web sites dedicated entirely to modding are extremely rare. Knowledge is shared primarily on discussion forums.

GTA modders have not (yet) produced any total conversions that would have amassed similar mod-specific fan communities as many *Half-Life* mods. However, the most extensive modding projects underway at the moment, such as the *Myriad Islands*, are community projects from the very start, with fans collaborating via dedicated discussion forums. These communities seem to be especially tight-knit.

The output of the *GTA* modding scene consists of skins, player models, vehicles (cars are by far the most popular form of user-created content), weapons, textures, map add-ons, code modifications and trainers. While the various custom content items are mainly shared and used separately, some fans have also gathered dozens of items from different authors into installable mod collections. The aforementioned community projects are also collections of sorts; although the general guidelines of the projects are discussed together, each participating fan creates their own contributions separately.

Although the authors often refer to the mod collections and community projects as total conversions, the changes they make to the original games are still rather superficial compared to *Half-Life* total conversions. Without access to the games’ source code *GTA* modders cannot create new game mechanics. What they can do is tweak and reassemble the existing ones using a proprietary scripting language. Gameplay-wise the most radical *GTA* modifications currently available are the two online multiplayer mods, though both are still too crude and unstable to wow casual gamers.

5.4.1. Notes

The distinguishing quality that makes *GTA* modding scene especially interesting is of course the fact that it has thrived without any developer involvement. All the tools and knowledge needed in modding has come from the fans themselves. This should be encouraging for game developers that appreciate the value of user-created content but are wary of the costs involved in developing editing tools and supporting mod communities. Few things must be remembered, however.

First of all, due to their huge commercial success, *GTAs* have an unusually large and devoted fan base. The fact that this

collective is able to nurture a self-sufficient mod culture could very well be the exception that proves the rule. (Of course, the vast majority of *GTAs* recent success can be contributed to the sales of PS2 games. The sales of the PC versions are likely to be good but nothing extraordinary.) Secondly, while the use of the RenderWare platform meant that Rockstar could not offer official support for modders, it is also a commonly used and well-documented technology. Though the documentation is not openly available, *GTA* modders have gotten hold of some this information, and therefore did not have to start from scratch when developing unofficial modding utilities. Finally, the previous installments in the *GTA* series - *Grand Theft Auto* (1996) and *GTA2* (1999) - were both very successful PC games and had an established modding scene of their own. Although these earlier 2d-games were technically from a different world, many of the pioneering *GTA3* and *Vice City* modders came from that scene.¹²⁷

It is of course possible, perhaps even probable, that the *GTA* modding scene has also benefited from the lack of official support. The fans have had to collaborate intensively to figure out the games' inner-workings, which undoubtedly has strengthened the communal ties (at least among the core members).

Though it is certainly impressive how far *GTA* modding scene has come on its own, even the most persistent fan community cannot totally overcome one issue related to the lack of official support: the un-mod-friendly game design. Many of the features that *GTA* modders would like to tweak have been hard-coded into the game executable. Not only is hacking the executable extremely difficult, it is also illegal (or more precisely it is illegal to distribute the hacked executable). There are ways around the problem: the *MultiTheftAuto* multiplayer mod, for example, makes temporary changes to the executable when it is loaded, thus making it unnecessary to distribute the *GTA* executable itself. Similar solution is used in the mod *GTA: Liberty City*, which replaces the whole game world of *Vice City* with that of *GTA*. To function, both *MTA* and *GTA: Liberty City* require that the player has the original games installed properly. As innovative as these sorts of workarounds are, they are also limited in their applicability.

The un-mod-friendly game design has confined *GTA* fans to rather superficial modding. Instead of being able to create completely new gaming experiences comparable to *Half-Life* total conversions, *GTA* modders must settle for tweaking and extending the original games. From a gamers perspective it seems unlikely that many of the *GTA* mods can sustain interest

¹²⁷ Some *GTA* sites still distribute mod files, tools and modding information for all the games in the series (<http://azz.gouranga.com/>).

for very long. They provide amusement for a while but are soon uninstalled and replaced by something else. And the often cumbersome installation process takes away some of that amusement. Overall it could be suggested that although the act of modding the *GTAs* may be extremely rewarding for the modders themselves, the final products have little attraction for gamers outside the community.

That being said, it must be remembered that the *GTA* modding scene is still evolving and the modding projects are constantly getting more ambitious. While the outcome of the *Myriad Island* community project might still not be a total conversion - it does not change the basic game mechanics - it could very well become a worthy unofficial episode in the popular series. Much of it depends on whether the community will also be able to produce interesting gameplay for the impressive new environment.

Whether there will ever be any "real" total conversions made for the *GTAs* remains to be seen. The next official title in the series is scheduled for release in the fall of 2004 with the PC version in all likelihood following the next spring. Perhaps the next PC version will be designed to be more mod-friendly. Maybe Rockstar will even release some official tools in conjunction. It's quite unlikely though, since the game will still use the licensed RenderWare platform. The bright side of this is that the current unofficial tools are most likely compatible with the sequel as well. Another year from now the game design will be even more thoroughly understood, and the unofficial tools will have evolved to the point that will let *GTA* modders tackle even more ambitious and innovative projects. On the other hand, if the core architecture of the next episode is radically different from its predecessors, modders will again have to start from square one.

6. Summary

As stated in the introductory chapter, the purpose of this preliminary study report was to provide the reader with an answer to the question of “what”: what kind of custom content do fans produce for their favorite games? As an added bonus, the question of “how” - i.e. how is the content created, shared and used - was also examined.

In an ideal situation these questions would have been approached by thoroughly investigating and documenting every game that has been actively modded (or at least one game from all game genres where modding has been a common practice). In reality however, given the limited resources, such an expansive approach would have resulted in a very cursory listing.

To find a better compromise between scope and depth, this study therefore concentrated on the modding scenes of just three games: *Half-Life*, *The Sims* and *Grand Theft Auto III / Vice City*. While these games all represent different genres and as such provide different viewpoints to the phenomenon of modding, it would be unwise to assume that three games alone could provide a complete account of the whole phenomenon. There are many other game genres - such as real-time strategy games, flight/train simulators and sports games - where modding is also common. Just as the modding scenes investigated in this study differ from each other, the games not represented have undoubtedly also spawned modding scenes with varying local nuances.

On the other hand, if one looks beyond details, the modding scenes investigated in this study also share many similar qualities. Based on casual observations of other modding scenes, these common qualities are, if not universal, nonetheless quite typical elsewhere in the modding subculture as well. The comparisons summary will highlight these common qualities while at the same time pointing out the major differences. The summary will also introduce some general concepts derived from the research that should be useful when trying to make sense of other modding scenes outside the scope of this study.

6.1. Comparing the cases

This summary will follow roughly the structure of the case studies. As such, the cases will be compared in three areas: 1) organization of the modding scene, 2) forms of custom content, and 3) availability of tools and support.

6.1.1. Organization of the modding scene

Perhaps the most defining common quality of the modding scenes investigated in this study is that they all seem to operate entirely online. No indication was found of any collaboration outside the internet. Though some game developers have organized real-world events for modders (e.g. Valve Mod Expos and Unreal University workshops), in general it could be argued that the current modding culture as a whole would not and could not exist without the internet. This should make the modding culture an interesting object of study for any research dealing with online cultures, not just for the projects related to gaming.

In each of the modding scenes studied here similar interactions take place among the fans: information sharing, content sharing, and collaboration. General socializing is obviously also an important function of any online communities, including modding communities. While the overall functions of the modding scenes are alike, there are some differences in the way in which the online communities have organized to serve these functions as the following table illustrates.

	Half-Life	The Sims	GTAs
Information sharing	Total conversion communities, dedicated modding communities, tutorials on personal sites	Discussion forums on general fan sites, tutorials on personal sites	Discussion forums on general fan sites, tutorials on personal sites
Content sharing	General game file archives (e.g. FilePlanet), total conversion specific archives, personal sites	Huge file archives on general fan sites, personal sites	File archives on general fan sites, personal sites, general file archives (e.g. FilePlanet)
Collaboration	Forums dedicated to total conversions, privately among mod teams, general support sites	Modding forums on general fan sites	Modding forums on general fan sites, dedicated forums for community projects, separate sites

Information sharing

Information sharing refers to the ways in which modding-related knowledge is distributed amongst modders. Of the three scenes investigated only the *Half-Life* scene has community sites dedicated entirely to modding. These sites (some of which are dedicated to a specific aspect of modding, such as mapping) offer tutorials and other modding-related information, and of course also host discussion forums. In *The Sims* and *GTA* scenes knowledge is shared primarily via discussion forums on general fan sites. Forums on bigger fan sites usually have separate sections for modding discussions. Besides community sites, in each scene fans have also set up personal sites where they can

offer tutorials on their own modding speciality (along with self-made custom content). The role of personal sites is most prominent in *The Sims* scene. A unique aspect of the *Half-Life* scene is the division to relatively autonomous total conversion communities. Much of the information sharing takes place via total conversion promo/fan sites. Some of the biggest mods even have separate sites for their own modding communities. General fan sites like Planet Half-Life, which report news from all the modding fronts, give the whole *Half-Life* modding scene a sense of unity. In *The Sims* and *GTA* scenes all fan sites are general and mainly report news related to the games or the contents of the site itself.

Content sharing

Content sharing refers to the ways in which custom content files are distributed. In the *Half-Life* scene general file archives such as FilePlanet are heavily used. The file sizes of total conversions are much too big for the amateur mod teams to host themselves. It is also convenient for the users, as the same files can be found on file archives across the globe. Smaller creations, such as custom maps and character models are shared via mod-specific file archives and personal sites. In *The Sims* and *GTA* scenes the main feature on most of the biggest fan sites is a huge custom content archive. General file archives are rarely used, though *GTA* modders use them to distribute major modifications. In *The Sims* scene custom content comes in small packages and is shared extensively via personal and small team sites. A notable feature in *The Sims* scene is that full access to the file archives and even custom content on personal sites often requires either a paid subscription or a donation.

Collaboration

Collaboration refers to the ways in which teamwork is organized in the modding scene. Most organized collaboration takes place in the *Half-Life* scene, where mod teams work together for months, even years to create total conversions. Much of this collaboration happens behind the scenes, though, via private discussion forums and personal correspondence. Once the total conversions are released, the fan community participates by testing, reporting bugs, suggesting improvements and also by creating new content: maps, skins, weapon models etc. Collaboration occurs also in smaller scale, as works in progress are posted to the total conversion forums for scrutiny and suggestions incorporated in revisions. Sometimes unfinished works (e.g. skinless models) are handed over for other modders to finish. This kind of small-scale collaboration is the only kind that takes place in *The Sims* scene. It is, for example, quite common that a skin artist uses custom meshes from another

modder as a basis for his/her works. In the *GTA* scene the modifications are becoming increasingly complex and, as a result, collaboration more extensive. Instead of being created by private mod teams, however, the most ambitious *GTA* mods - such as the *Myriad Islands* - are open community projects from the start. There are also few more private *GTA* mod projects, the multiplayer mod *MultiTheftAuto* being a prime example.

6.1.2. Forms of custom content

A large portion of this report was spent describing in detail the various custom content types produced in each of the scenes. This was done to ensure an adequate representation of the diversity in fan producers' output in each scene. However, despite the local differences, it is possible to recognize certain main types of custom content that are common for all the scenes investigated. While a more generalized taxonomy will inevitably lose some of the nuances, it should also be considerably more useful for subsequent studies.

There are naturally several ways in which the custom content forms could be classified. The taxonomy presented here starts with two main categories: audio-visual modifications and mechanics modifications. Audio-visual modifications change the way the game world looks and sounds, while mechanics modifications change the way it functions. The two categories can be further expanded into the subcategories as follows:

Audio-visual modifications	Mechanics modifications
Gameplay environments Custom characters Custom items Interface elements Sounds	Scripting Programming

Of course there are plenty of user creations that are combinations of both audio-visual and mechanical modifications. Maps, for example, define the gameplay arena but can also include scripted functionality, and total conversions usually contain all custom content types. With that in mind, the following table illustrates how the custom content types produced for the investigated games fit into the categories:

	Half-Life	The Sims	GTAs
Audio-visual			
Gameplay environments	Maps, prefabs, map models, textures, sprites, logos	Houses/lots, walls, floors, roofs, objects	Map add-ons, textures
Custom characters	Models and skins	Head and body meshes, skins	Models (<i>GTA3</i>) and skins
Custom items	Weapon models and	Objects, hacked	Vehicle models and

	skins	objects	textures, weapon models
Interface elements	Sprites	-	(Interface) textures, load screens
Sound	Effects, dialogue, ambience	-	Radio stations
Mechanics			
Scripting	Map scripts, character/weapon properties	Hacked objects	Code modifications
Programming	Total conversions, bots	-	Loader applications for some major modifications
	Half-Life	The Sims	GTAs

Based on observations of other modding scenes the taxonomy presented above seems to be generalized enough to be universally applicable. As the table demonstrates, examples of most content types can be found for each investigated game. The major differences are in the emphasis or popularity of the content types. This emphasis can be derived quite directly from the game design itself, though the developers' official support agenda is also a contributing factor.

Audio-visual modifications

Gameplay environments

Modifications of the gameplay environment change the visual appearance of the game world, some more extensively than others. The three games examined all have quite different gameplay environments. *Half-Life*, with its level-based design, is closest to the "norm". As is generally the case with online shooters, custom maps are the most popular form of custom content for *Half-Life*. Many other forms of custom content - such as textures, prefabs and scripting - are also tied to mapping. Each map in an online game is self-contained, so new maps can be created and added to server rotations practically infinitely. Furthermore, new maps are also required to keep the games interesting year after year. It is not surprising that Valve has decided to support mapping by bundling the official level editor with the game. In *The Sims* the gameplay environments - lots - are created with very simple in-game tools but can be exported and shared as separate files. Downloaded lots are easy to install and can be further modified as needed. Although the lots can be built and furnished from the pre-provided elements, the thousands of custom walls, floors and objects available on the web greatly expand the possibilities. Nevertheless, compared to *Half-Life* mapping, *The Sims* "lotting" is much more restricted. In the *GTAs*, because of the games' unique level-less design, gameplay environment modifications are

mostly localized changes ranging from simple texture replacements to map add-ons such as stunt park islands. In the Myriad Island community project the entire gameworld is fan-made. The lack of official tools has undoubtedly seriously limited the popularity of *GTA* mapping.

Custom characters

Custom characters can be either player characters or NPCs. In each of the games characters are made of 3D models/meshes wrapped into skin bitmaps. *Half-Life* modders usually create and share complete characters, where both the model and the skin(s) are custom made. Original models from total conversions are often used as a starting point for tweaking. In *The Sims* and *GTA* scenes, on the other hand, it is more common to share custom skins separately. Dressing up the virtual people is of course a central element of *The Sims* gameplay, and the massive popularity of skinning understandable. Custom skins range from celebrities to comic book characters, from Versace fall collections to Vikings, enabling a huge variety of gameplay scenarios. Creation and installation of custom skins is also extensively supported officially (see below). Custom body and head meshes for *The Sims* are little more rare, but valued by skimmers as they allow more variety in clothing and hairstyles. For the *GTAs*, skinning is the only officially supported form of customisation. While custom skins are abundant, custom models are rare. The character animations in both *The Sims* and *GTAs* are non-editable, which greatly limits the freedom of customisation. In *Half-Life* character animations are fully editable, and the only limit to the look of custom models is imagination.

Custom items

The category of custom items is somewhat vague, referring to any user-created in-game elements that are not static parts of the gameplay environment or characters. *Half-Life* total conversions can have a variety of usable items (health packs etc.), but generally only weapon models are further customized by the fans. Weapon modelling is extremely popular among *Half-Life* modders - more so in certain total conversion communities - which is not surprising considering the genre. In *The Sims* all custom items are objects, though the objects that are used as building blocks perhaps fit better under the gameplay environment category. Most objects, however, have various interactions attached to them. Despite the puzzling lack of official support, object making has become very popular among *The Sims* fans. *GTAs'* custom items are vehicles and weapons. Of these the former is overwhelmingly more common and easily the most popular form of *GTA* customisation. Most of

the time in the games is spent in/with cars, so it is not surprising - especially since car modelling in general is very popular among 3D hobbyists.

Interface elements

Mod teams usually create customized interfaces for *Half-Life* total conversions to fit their reworked game systems. These interfaces include elements such as weapon icons and health bars and are constructed of special purpose bitmaps called sprites. Sprites are very easy to edit further, and various fan made custom sets are available for most total conversions. While in the *GTA* scene the term "sprite" is not used, interface texture sets with the exact same functionality are available. *GTA* fans have also created custom load screens for the games. *The Sims* interface has not been customized.

Sounds

Sound modding is overall the least popular area of customisation. In the *Half-Life* scene total conversions usually do have a wide range of new sounds, though, and custom sounds can also be attached to fan made maps. Fans have also created various sound packs mostly for weapons. The sounds are usually ripped directly from other games. In *The Sims* and *GTA* scenes custom sounds are practically non-existent. *The Sims* allows users to change the sounds of radio stations and television shows very easily, but the large file sizes and copyright issues discourage distribution. The same is true with *GTAs* car radios, though some custom stations are available.

Mechanics modifications

Scripting

Scripting takes advantage of the functions defined in the existing program code. *Half-Life* modders use scripting mainly in mapping, e.g. to create pre-defined action sequences for AI controlled monsters. Multiplayer maps rarely have need for scripted sequences, so scripting is mostly used in single-player maps. *The Sims* fans use scripting to create so-called hacked objects. Hacked objects have modified functionality and often trigger customized animation sequences. In the *GTA* scene scripting is used very extensively to manipulate the game mechanics and to create whole new gameplay objectives. The big problem, however, is that only one script modification can be used simultaneously.

Programming

Of the three games examined, *Half-Life* is the only one that allows actual reprogramming of the game system. Part of the source code was released in the SDK. With generic C++ editor modders can create new functionality and basically completely rewrite the game system. Total conversions are always based on a modified source code, though the extent of modification varies greatly. *Half-Life* fans have also programmed “bots” - AI controlled enemies and teammates - for the total conversions. Bots are usually distributed as separate sub-programs. Reprogramming the GTAs is not officially supported, and distribution of modified game code forbidden. However, some of the major modifications - e.g. the multiplayer mods - use external loader applications that temporarily modify the game system as the game is loaded.

6.1.3. Tools and support

While the focus of this report was on the various forms of fan-made game content (the “what” of modding), the availability of modding tools and modes of support (or the “how”) in each scene was also addressed. This area will be further explored in subsequent studies (see below), but already it is possible to suggest certain common and diverging features in the investigated scenes. As with custom game content, casual observations indicate that similar features can be recognized in other modding scenes as well.

The following table summarizes the findings on tools and support. On both sections there are three subcategories: “official” (tools/support provided by the developer); “unofficial” (tools/support from the fan community); and “other” (tools/support from a third party source, i.e. generic graphics applications).

	Half-Life	The Sims	GTAs
Tools			
Official	Level editor bundled with the game, some command line utilities (for format conversions etc.) in the SDK	Several decidedly user-friendly applications mostly for character customisation; also for some format conversions and limited content management. In-game house building mode.	-
Unofficial	Game data file utilities, better compilers for the level editor, custom model and skin viewer/exporter-importer, texture editor	Dozens of tools to complement and substitute the feature-short official ones. Object editing/hacking and content management tools esp. important.	Game data file utilities, map editor, collision file editors, script editors/compilers, texture editors. Custom content management/installation utilities. Often many alternatives for the same

			task.
Other	Generic image and audio editors; 3D modelling tools (notably Milkshape, developed initially for HL modelling); C++ programming tools and compilers for coding.	Generic image editors; 3D modelling tools.	Generic image editors; 3D modelling tools.
Support			
Official	Hammer level editor and other official utilities well-documented, VERC community for modders, annual Valve Mod Expos hosted to promote best total conversions.	Adequate documentation on official tools. Custom content exchange via the official fan community site. Some custom content related discussions on general support forums.	-
Unofficial	Many dedicated modding communities, most focusing on mapping. Mod-specific editing forums on total conversion sites. Tutorials on personal fan sites. General fan sites report news from the modding scene.	Dedicated editing forums on general fan sites. Modding-related Yahoo groups. Tutorials on personal fan sites. Promotion with affiliate banners.	Dedicated editing forums on general fan sites; GTAForums esp. important.
Other	General image editing and 3D modelling resources and tool-specific sites. C++ programming sites.	General image editing and 3D modelling resources and tool-specific sites.	General image editing and 3D modelling resources and tool-specific sites.
	Half-Life	The Sims	GTAs

Tools

Official

In regard of official tools the *Half-Life* scene is the most representative of modding scenes in general. A level editor is the one “in-house” tool that developers most often provide, and as with *Half-Life*’s Hammer, the editor is usually bundled with the game. This common practice is quite rational, since constant flow of custom maps is pivotal for the longevity of online games. *The Sims* fans have been treated to numerous official tools, most of which are related to character customisation. Compared to Valve, Maxis has targeted a very different demographic with its decidedly simple utilities – too simple for most modders.

Unofficial

Since *Half-Life* was built on a *Quake* engine, many of the unofficial tools created for *Quake* were readily compatible. Only few tools - such as the Half-Life Model Viewer - have emerged from the *Half-Life* community itself. Overall, *Half-Life* modders make do with relatively small tool set. *The Sims* fans, on the other hand, can take their pick from dozens of fan-made utilities. Most of these are of course designed for the same tasks. Easily the most important unofficial tool for *The Sims* scene has been the Transmogrifier, which enables the creation of custom objects - something completely overlooked by Maxis. Even for the tasks where official tools are available, however, *The Sims* modders tend to prefer unofficial alternatives because of their versatility. *GTA* fans have had to build all tools for themselves, and through reverse-engineering have produced a line-up almost as extensive as the one for *The Sims*. For most modding needs there are usually several alternatives available, but there are also still areas that fans need to figure out better to be able to produce tools. It is noteworthy that in both *The Sims* and the *GTA* scene majority of unofficial tools come from only a handful of dedicated toolmakers. Some of these "toolsmiths" have alone produced more than ten separate utilities. In the *GTA* scene the toolmakers collaborate extensively via GTAForums community site.

Other

In addition to game-specific modding tools modders always also need various generic applications, such as image editors, modelling tools, and program code compilers. These applications can be very expensive - a popular modelling tool 3ds max retails for around 5000 euros, while the image editor Photoshop costs 1000 euros. Based on the forum discussions and tutorials, these are both commonly used among modders. It is unclear how legitimately these tools are used, though. One could assume that software piracy is quite crucial for the livelihood of the modding scenes. On the other hand there are some affordable alternatives available. For example the 3D modelling tools Milkshape and Zmodeler are both designed for modders and cost only 20-30 euros. Game developers are also increasingly forming partnerships with 3D software companies. *Half-Life 2* modders can already download a special version of the SoftImage's XSI tool for free. The tool has all the features of a full version, but can only save files in the *Half-Life 2* model format.

Support

Official

Official modding support can take many forms: the developer can provide modding tools and tutorials, host modding communities, participate in forum discussion, organize workshops, offer hosting services, etc. Of course, a mod-friendly game design itself is one of the most important forms of support. Except for hosting services, Valve has provided all these support forms for the *Half-Life* modders. Additionally Valve has promoted the best fan creations at annual Mod Expo events. Maxis has also provided quite extensive support for The Sims fans, including numerous tools (with adequate documentation) and a custom content exchange service. Surprisingly, though, the official community site offers very little modding-related information. *GTA* fans have not received any explicit official support, but the game architecture is quite amenable for certain kinds of customisation, as many properties and attributes are defined in plain text files.

Unofficial

While the official VERC community provides more than adequate support *Half-Life* modders with its tutorials, discussion forums and knowledge databases, this has not stopped fans from setting up numerous unofficial communities. Most of these have specialized in certain area of modding, such as mapping or programming. A defining feature in the *Half-Life* scene is of course the proliferation of very active total conversion fan communities, which have a central role in the supportive duties. General fan sites, such as Planet *Half-Life*, which report news from the whole scene, support the mod teams by promoting their works. As noted above, the official Sims community offers very little information for the modders. As there are no sites dedicated entirely to *The Sims* modding¹²⁸, modding knowledge is primarily shared via discussion forums on general fan sites. Most sites have separate sections for modding discussions. Although *The Sims* fan sites rarely report news that are not related to the site itself, prominently displayed affiliate banners form an effective promotional network. The unofficial support practices in the *GTA* scene resemble *The Sims* scene. There are no “pure” modding communities, but bigger fan sites generally have discussion forums dedicated to modding. Affiliate networks are also common. In each of the investigated scenes there are plenty of personal sites where fans share tutorials on their modding specialties.

Other

¹²⁸ There is, however, such a site for *The Sims* 2: <http://www.modthesims2.com/>.

It is of course impossible to list all the possible forms of support modders may receive outside the modding communities. A class on human anatomy, for example, can be extremely valuable for a character modeller. On the internet modders can find vast amounts of technical assistance on the support communities dedicated to specific applications. Most applications have at least one support forum on the official product site, but popular tools such as Photoshop and 3ds max also have numerous unofficial information resources. While these sites may not have information related directly to modding, they will teach modders the basics and demonstrate more advanced tricks that allow them to take full advantage of the tools' features.

6.2. Further research objectives

The research documented in this report has been decidedly descriptive and detached in nature. The aim has been to serve as an objective introduction to the diversity of user-created content and to lay groundwork for further, more analytical and focused studies into the modding phenomenon. Obviously the perspective most sorely missing from the report at hand is that of the modders themselves. To elaborate:

- What makes some people want to modify their games? Who are these people?
- Why do modders prefer certain forms of custom content? How do they decide which game(s) they want to modify?
- What is the meaning of the various online networks of modding scenes for individual modders? What kind of roles and practices develop in the modding communities?
- What kind of support do modders expect from the developers? What is the relationship like overall?

To some extent these questions could be answered by close reading the numerous modding related discussion forums, but for more in-depth insights, surveys and personal interviews with modders should be conducted.

Of course, the investigation of the modding culture would be quite lopsided if the focus was solely on modders. It is also crucial to understand the users of the mods. Why do gamers install modifications to their games? What forums of interaction exist between modders and gamers? How do various forms of custom content affect the gameplay experience?

Although game developers have been allowed to relate their views on fan production quite extensively in the games media (see Edge 2003, Equip PC 2003), it would certainly be worthwhile to conduct in-depth interviews with industry

representatives from an academic perspective. If nothing else, this approach could provide a welcome counterpoint to the leftist slant of the many current socio-economic accounts on modding culture (and participatory fan cultures in general).

Finally, the modified games themselves should not be overlooked. Although the ethnographic approach can reveal the personal meanings attached to mods and modding, there are certain issues that can only be understood if the games at the centre of the whole phenomenon are examined. For instance, what kind of themes, mechanics and other design solutions in games inspire (certain kind of) fan productivity?

The multi-edged approach outlined above will not only produce a thorough understanding of the modding phenomenon itself, but it should also provide a wealth of practical information that can be applied to developing and supporting the modding scenes of the future.

7. References

- Banks, J. (2003) "Negotiating Participatory Culture in the New Media Environment: Auran and the Trainz Online Community - An (Im)possible Relation". *Fine Art Forum*, Vol 17, Issue 8 (August 2003).
<<http://hypertext.rmit.edu.au/dac/papers/Banks.pdf>>
- Himanen, P. (2001). *Hakkerietiikka ja informaation henki*. WSOY, Helsinki. [Note: English translation available: *The Hacker Ethic and the Spirit of the Information Age*. Random House, New York. (2000)]
- Jenkins, H. (1992). *Textual Poachers. Television Fans And Participatory Culture*. Routledge, New York & London.
- Jenkins, H. (2002). "Interactive Audiences?" In Harries, D. (ed.) *The New Media Book*. British Film Institute, London.
- Jenkins, H. (2003). "Quentin Tarantino's Star Wars?: Digital Cinema, Media Convergence, and Participatory Culture." <<http://web.mit.edu/21fms/www/faculty/henry3/starwars.html>>.
- Jeppesen, L. and Molin, M. (2003) "Consumers as Co-Developers. Learning and Innovation Outside the Firm". Department of Industrial Economics and Strategy, Copenhagen Business School, Denmark.
- Kücklich, J. (2005). "Precarious Playbour. Modders and the Digital Games Industry". *Fibreculture Journal*. Issue 5. <<http://journal.fibreculture.org/issue5/kucklich.html>>
- Laukkanen, Tero (2004). "Fanituottajuuden kulta-aika. Digitaalinen mediateknologia ja osallistuva kulttuuri". In *Lähikuva* 4/2003 (January 2004). pp. 36-47.
- Levy, Steven (1994). *Hackers. Heroes of the Computer Revolution*. Penguin Books, New York.
- Mactavish, A. (2003). "Game Mod(ifying) Theory: The Cultural Contradictions of Computer Game modding", delivered at Power Up: Computer Games, Ideology and Play, Bristol UK, July 14-15, 2003.
- Sotamaa, O. (2003). "Computer Game Modding, Intermediality and Participatory Culture." Presented at "New Media? New Theories? New Methods?" PhD course organised by The Nordic network "Innovating Media and Communication Research", Sonderborg, Denmark, December 1-5, 2003. <http://www.imv.au.dk/eng/academic/pdf_files/Sotamaa.pdf>
- Wright, Breidenbach & Boria (2002). *Creative Player Actions in FPS Online Video Games*. Game Studies. <<http://www.gamestudies.org/0202/wright/>>

Non-academic

- Antoniades, Alexander (1994). 'The Doom of Doom', *Game Developer*, June 1994. pp. 39-41.
- Au, W. J. (2002). "Triumph of the mod", *Salon.com*.
<<http://www.salon.com/tech/feature/2002/04/16/modding/>>
- Edge [anonymous writer] (2003). "The Modern Age", *Edge*, Issue #126, (September 2003). pp. 58-67.
- Equip PC [anonymous writer] (2003). "Building Blocks", *Equip PC*. pp. 64-75.
- Kushner, D. (2003). *Masters of Doom. How two guys created an empire and transformed pop culture*. Random House, New York.